



2265 CREACIÓ D'UN FRAMEWORK PER APLICACIONS WEB AMB LA PLATAFORMA .NET

Memòria del projecte
d'Enginyeria en Informàtica
realitzat per Sergi Pedra
Pagès i dirigit per Josep
Maria Sánchez Castelló

Bellaterra, Setembre de 2010



El sotasignat, Josep Maria Sánchez Castelló
Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Sergi Pedra Pagès.

I per tal que consti firma la present.

Signat:

Bellaterra, Setembre de 2010

El sotasignat,
de l'empresa, Diputació de Lleida,

CERTIFICA

Que el treball a què correspon aquesta memòria ha estat realitzat en l'empresa sota la seva supervisió mitjançant conveni
firmat amb la Universitat Autònoma de Barcelona.

Així mateix, l'empresa en té coneixement i dóna el vist-i-plau al contingut que es detalla en aquesta memòria.

Signat:

Lleida, Setembre del 2010

Índex

1	Introducció	1
1.1	Motivacions	1
1.2	Objectius	2
1.3	Organització de la memòria	2
1.4	Informe Previ	3
1.4.1	Objectiu/s del projecte	4
1.4.2	Breu introducció a l'estat de l'art del tema proposat	4
1.4.3	Estudi de viabilitat del projecte	5
1.4.4	Planificació temporal del projecte	6
1.4.5	Altres comentaris	7
2	Descripció	9
2.1	Fonaments teòrics	9
2.2	Requeriments	15
2.3	Planificació i fases	17
2.4	Disseny	22
2.5	Implementació	29
2.5.1	Visual Basic .NET	29
2.5.2	JavaScript:	49
2.5.3	CSS	54
2.6	Cas pràctic	56
3	Conclusions	61

A	Eina de gestió d'una escola	65
B	Servei Web	71
C	Aplicació Web utilitzant la llibreria d'empresa X	75
	Bibliografia	79

Índex de figures

1.1	Diagrama Gantt de l'informe previ	6
2.1	Estructura del Framework .NET	10
2.2	Estructura interna del CLR	11
2.3	Diagrama bàsic de la Biblioteca de Classes Bàsic	12
2.4	Entorn de desenvolupament del Visual Studio	14
2.5	Diagrama Gantt	17
2.6	Diagrama de seqüència del framework	23
2.7	Diagrama de classes dels controls orientats a la capa de presentació	25
2.8	Diagrama de classes dels controls enllaçats a dades	28
2.9	Diagrama de seqüència del constructor de la classe Data	31
2.10	Diagrama de seqüència del constructor de la classe Hora	33
2.11	Diagrama de seqüència del constructor de la classe Document Identificatiu	35
2.12	Diagrama de seqüència de primer constructor de la classe LabelText	38
2.13	Diagrama de seqüència del segon constructor de la classe LabelText	39
2.14	Diagrama de seqüència del constructor de la classe numCompte .	42
2.15	Diagrama de seqüència del constructor de la classe PanelForm . .	45
2.16	Diagrama de seqüència del constructor de la classe GridViewForm	47
2.17	Disseny de la base de dades de l'aplicació corresponent al cas pràctic	56
2.18	Captura de pantalla del menú de l'aplicació desenvolupada en el cas pràctic	57
2.19	Captura de pantalla de la pàgina corresponent a l'operació d'afegir un treballador de l'aplicació desenvolupada en el cas pràctic . . .	58

2.20	Captura de pantalla de la pàgina corresponent a l'operació d'afegir una absència de l'aplicació desenvolupada en el cas pràctic	58
2.21	Captura de pantalla de la pàgina corresponent a l'operació de consultar absències de l'aplicació desenvolupada en el cas pràctic	59
2.22	Captura de pantalla del Pop-Up que es desplega quan es selecciona el treballador del qual es vol consultar les absències	59
A.1	Captura de pantalla on un usuari pot elegir entre el perfil d'administrador o usuari normal	66
A.2	Captura de pantalla on demana que l'administrador s'identifiqui mitjançant un nom i una contrasenya	67
A.3	Captura de pantalla on un administrador pot elegir l'operació a realitzar	67
A.4	Captura de pantalla que permet afegir un alumne a la base de dades	68
A.5	Captura de pantalla que permet eliminar un alumne existent a la base de dades	68
A.6	Captura de pantalla que permet posar notes d'un alumne existent a la base de dades	69
A.7	Captura de pantalla que permet modificar algun dels paràmetres d'un alumne existent a la base de dades	69
A.8	Captura de pantalla que mostra l'expedient acadèmic d'un alumne en concret	70
B.1	Captura de pantalla que mostra els diferents mètodes que es poden invocar	72
B.2	Captura de pantalla que mostra l'operació de sumar	72
B.3	Captura de pantalla que mostra l'operació de restar	73
B.4	Captura de pantalla que mostra l'operació de multiplicar	73
B.5	Captura de pantalla que mostra l'operació de dividir	74
B.6	Captura de pantalla dels resultats de les diferents operacions retornades per el servidor Web	74
C.1	Captura de pantalla que mostra la pàgina d'inici del portal	76

C.2	Captura de pantalla de la pàgina que permet donar d'alta un nou usuari	76
C.3	Captura de pantalla de la pàgina del menú d'inici	77
C.4	Captura de pantalla de les dades retornades de l'operació consultar	77
C.5	Captura de pantalla que mostra la pàgina per afegir dades	77
C.6	Captura de pantalla que mostra la pàgina per eliminar dades . . .	78
C.7	Captura de pantalla que mostra la pàgina per modificar dades . . .	78

Capítol 1

Introducció

En els temps que vivim, moltes de les aplicacions de gestió que existeixen, són aplicacions Web, ja que són fàcilment accessibles pels usuaris, l'únic software que es necessita és un navegador Web i és més fàcil de que sigui multiplataforma. És per això que es vol desenvolupar un framework per fer aplicacions Web .NET. Tot seguit s'expliquen les motivacions i els objectius.

1.1 Motivacions

Avui en dia, Internet ha provocat una revolució en el món de la comunicació, ja que és una tecnologia barata d'explotar, té poques limitacions i és fàcilment accessible pels usuaris. Coneixedors d'aquesta nova revolució, hi ha hagut empreses, com per exemple Microsoft i Sun Microsystems que han creat noves plataformes de desenvolupament de software, que tenen en compte aquestes noves necessitats.

La nova plataforma de desenvolupament de software de Microsoft, s'anomena .NET; és una plataforma que ofereix la possibilitat que el software pugui ser executat en un sistema de manera independent al hardware (per exemple a una PDA o als aparells mòbils) i a més a més permet un ràpid desenvolupament d'aplicacions. La seva proposta és oferir d'una manera ràpida, econòmica i segura, el desenvolupament d'aplicacions; fent possible una ràpida i àgil integració entre

empreses i alhora un accés més simple i universal a tot tipus d'informació.

Hi ha moltes empreses que utilitzen la plataforma .NET per desenvolupar aplicacions de gestió Web, una d'aquestes és l'empresa per la qual es fa aquest projecte. Un dels problemes que es troben és la falta de temps a l'hora de fer front als diferents projectes que hi ha pendants de realitzar; degut això volen implementar un conjunt d'eines que els hi faciliti la programació d'aplicacions Web.

1.2 Objectius

L'objectiu del projecte, consisteix en implementar un framework perquè sigui menys costos el desenvolupament d'aplicacions Web .NET. A continuació es poden veure els subobjectius del projecte.

- Coneixement exhaustiu de la plataforma de desenvolupament d'aplicacions .NET.
- Definició dels elements necessaris per codificar aplicacions de gestió i optimitzar el temps de desenvolupament en la capa de presentació.
- Definició d'eines que permetin facilitar l'ús dels controls enllaçats a dades dins de l'entorn de desenvolupament.
- Realitzar una aplicació pràctica amb les noves eines desenvolupades amb la finalitat de comprovar el correcte funcionament de les eines i la millora en el temps necessitat per desenvolupar l'aplicació.

1.3 Organització de la memòria

Aquesta memòria està formada per tres capítols: l'introducció, la descripció i les conclusions.

- Introducció: aquest capítol conté quatre apartats.
 - Motivacions: es fa una breu introducció al problema.
 - Objectius: s'enumera els diferents objectius que s'ha d'assolir en aquest projecte.
 - Organització de la memòria: s'explica com està organitzada la memòria.
 - Informe previ: s'exposa l'informe previ que es va entregar al mes de Febrer.
- Descripció: aquest capítol conté cinc apartats.
 - Fonaments teòrics: s'explica quins són els coneixements necessaris per tal de poder realitzar el projecte.
 - Requeriments: s'enumera els requeriments necessaris per tal de desenvolupar el projecte.
 - Planificació i fases: es parla de com s'ha planificat el projecte i es fa una breu explicació de cada una de les fases que el componen.
 - Disseny: s'explica i es justifica com s'ha dissenyat el projecte.
 - Implementació: s'explica i es justifica com s'ha implementat el projecte.
 - Cas pràctic: s'explica com s'ha fet una petita aplicació utilitzant les eines del framework per tal de fer una demostració del funcionament de les noves eines.
- Conclusions: es comenten les conclusions extretes del projecte un cop finalitzat.

1.4 Informe Previ

Al Febrer es va entregar un informe previ; on hi constaven els diferents apartats: els objectius del projecte, un informe de viabilitat, la planificació temporal del

projecte, etc. A continuació s'exposa l'informe que es va entregar al Febrer.

1.4.1 Objectiu/s del projecte

- Coneixement exhaustiu de la plataforma de desenvolupament d'aplicacions .NET.
- Definició dels elements necessaris per codificar aplicacions de gestió i optimitzar el temps de desenvolupament en la capa de presentació.
- Definició d'eines que permetin facilitar el model de dades dins de l'entorn de desenvolupament.
- Realitzar una aplicació pràctica consistent en el desenvolupament d'un procés de pagament telemàtic.

1.4.2 Breu introducció a l'estat de l'art del tema proposat

Avui en dia moltes empreses s'aprofiten de que Internet és una tecnologia de comunicació molt barata d'explotar, que té poques limitacions i es fàcilment accessible pels usuaris, és per això que moltes aplicacions de gestió es programen pel World Wide Web.

Coneixedors d'aquest nou boom, empreses com Sun Microsystems i Microsoft han creat plataformes de desenvolupament de software que tenen en compte aquestes necessitats. Per exemple, la plataforma creada per Microsoft s'anomena .NET.

.NET és una de les tecnologies més utilitzada per les empreses; però també es troben en limitacions de temps a l'hora de poder fer front a tots els projectes que tenen plantejats desenvolupar, és per això que algunes empreses volen definir noves eines per tal de tenir una optimització de temps i una facilitació en els models de dades.

1.4.3 Estudi de viabilitat del projecte

Especificacions del sistema i de les aplicacions necessàries

- Microsoft .NET framework.
- Microsoft Visual Studio 2005.
- Microsoft Sql Server 2007.
- NHibernate.

Viabilitat tècnica

Tècnicament és viable degut a que és possible portar-lo a terme satisfactòriament ja que es treballarà amb el programari adient. Aquest programari ocupa poca memòria si tenim en compte els grans tamany dels discs durs que hi ha avui en dia. Necessita 4GB de memòria RAM perquè funcioni d'una forma fluida tot i que amb 2GB també ens funcionaria.

També es tindrà en compte que un cop finalitzat es pugui afegir millores per tal que no es quedi desfasat i d'aquesta manera sigui durable.

Viabilitat econòmica

- Microsoft .NET framework, ja està inclòs amb moltes versions del Windows, però tot i així es pot descarregar de forma gratuïta.
- Microsoft Visual Studio 2005, es pot obtenir de la MSDN Academic Alliance o el pot proporcionar l'empresa.
- Microsoft Sql Server 2007, el pot proporcionar l'empresa.
- NHibernate, és software lliure.

Viabilitat legal

No té cap restricció legal.

Conclusions

És viable dur a terme el projecte.

1.4.4 Planificació temporal del projecte

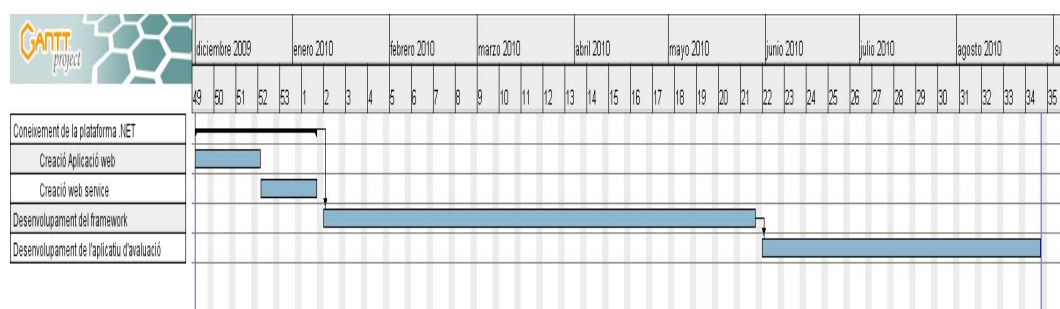


Figura 1.1: Diagrama Gantt de l'informe previ

- Coneixement de la plataforma .NET: aquesta part serà la que ens permetrà adquirir un coneixement de la plataforma de desenvolupament .NET. Constarà de dues fases, la primera serà la creació d'un senzill aplicatiu Web per tal de conèixer la sintaxi. La segona fase consistirà amb la creació d'un senzill Web service.
- Desenvolupament del framework: aquesta és l'etapa més important del projecte i la que necessitarà més temps; consistirà en definir noves eines i nous elements per codificar aplicacions de gestió i facilitar el model dades. Ens permetrà assolir el segon i el tercer objectiu esmentat al primer apartat.
- Desenvolupament de l'aplicatiu d'avaluació: l'última etapa, serà una etapa de testeig: consistirà en el desenvolupament d'un procés de pagament telemàtic utilitzant les eines i els elements creats en l'etapa anterior; per tal de comprovar el seu correcte funcionament.

1.4.5 Altres comentaris

Per dur a terme aquest projecte farem servir diversos llenguatges de programació. S'haurà de tenir un coneixement exhaustiu de la sintaxis de l'ASP.NET i els seus components; també de C#. A part dels llenguatges esmentats anteriorment, utilitzarem el llenguatge JavaScript. No podem oblidar l'importància de l'ús del NHibernate, un programa que ens permet fer un mapeig objecte-relacional; però haurem de saber com configurar-lo i com usar aquesta eina. S'haurà de realitzar consultes de Microsoft SQL Server, per tant és de vital importància conèixer la sintaxis d'aquestes consultes.

Capítol 2

Descripció

En aquest capítol s'explica com s'ha dut a terme el projecte. Primer es parla dels fonaments teòrics i els requeriments necessaris, després es pot veure com s'ha planificat el projecte i una petita explicació de cada una de les fases, també es parla de com s'ha dissenyat i implementat. Finalment s'ha realitzat un aplicació de gestió Web per comprovar el funcionament de les noves eines.

2.1 Fonaments teòrics

En aquest apartat, es parla dels coneixements necessaris per tal de realitzar el projecte i es fa una breu explicació de cada un d'ells. Els coneixements necessaris són els que s'esmenten a continuació.

- **Framework .NET:** és un component de software que pot ser afegit al sistema operatiu Windows. Conté un extens conjunt de solucions predefinides per la programació d'aplicacions i pretén ser utilitzada per la majoria de les aplicacions creades per la plataforma Windows. A la figura 2.1 es pot observar l'estructura del Framework .NET.

Els principals components de la plataforma són:

- Common Language Runtime (CLR): el CLR és el nucli del Framework de .NET. És el responsable de compilar el codi font de qualsevol

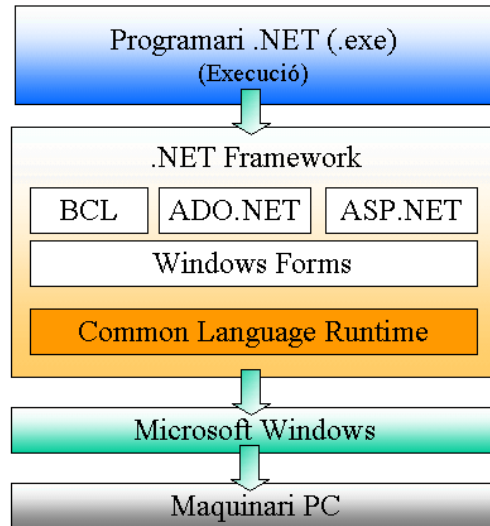


Figura 2.1: Estructura del Framework .NET

dels llenguatges suportats per .NET, com per exemple el Visual Basic .NET o el C#, en un codi entremig, anomenat el MSIL¹, el MSIL és similar al BYTECODE de Java. Per generar aquest codi, es necessàri unes regles per crear el codi MSIL compatible amb el CLR, per tant, el compilador es basa en les especificacions CLS².

Es necessita un segon pas per executar-se, un compilador, el compilador utilitzat és el JIT³, que és l'encarregat de generar el codi màquina que s'executa a la plataforma del client. D'aquesta forma s'aconsegueix l'independència de la plataforma hardware.

Si es vol veure l'estructura interna del CLR, mirar la figura 2.2.

- La Biblioteca de Clases Base o BCL: és la responsable de controlar la major part de les operacions bàsiques que es troben involucrades en el

¹Microsoft Intermediate Language

²Common Language Specification

³Just-In-Time



Figura 2.2: Estructura interna del CLR

desenvolupament d'aplicacions, algunes de les operacions són:

- * Interacció amb dispositius perifèrics.
- * Control de dades (ADO.NET).
- * Transmissió i recepció de dades per diferents mitjans (XML, TCP/IP).
- * Administració de components Web que corrent tant en el servidor com en el client (ASP.NET).
- * Control del sistema de finestres.
- * Eines de seguretat i integració amb la seguretat del sistema operatiu.
- * Controls de cadenes de caràcters i expressions regulars.
- * Operacions aritmètiques.
- * Interacció amb el API Win32 o Windows API.

Per observar el diagrama bàsic de la Biblioteca de Classes Bàsic veure figura 2.3.

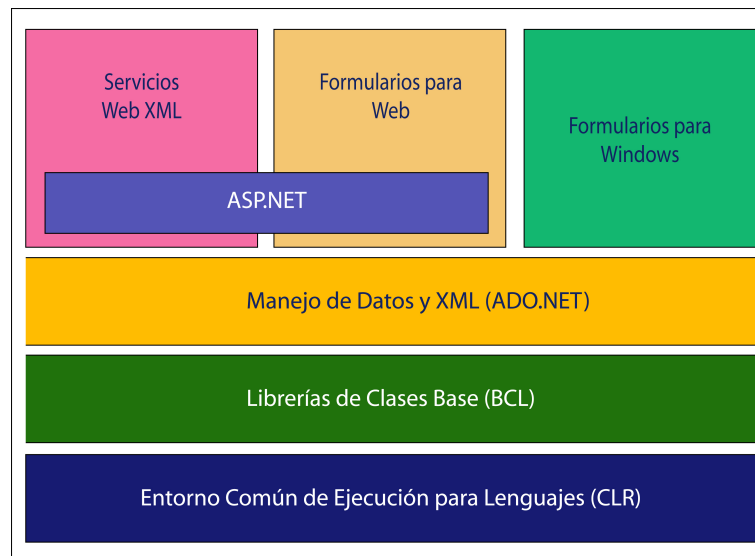


Figura 2.3: Diagrama bàsic de la Biblioteca de Classes Bàsic

- El conjunt dels llenguatges de programació: alguns dels llenguatges suportats per el Framework .NET són:
 - * C#.
 - * Visual Basic .NET.
 - * C++.
 - * Python.

- **ASP.NET:** és un framework per aplicacions Web desenvolupat per Microsoft, és la successora del ASP⁴. Els programadors poden escriure codi ASP.NET utilitzant qualsevol llenguatge admès per el Framework .NET, ja que està construït sobre el Common Language Runtime. El ASP.NET és utilitzat per programadors amb l'objectiu de construir llocs Web dinàmics, aplicacions Web i serveis Web XML.

Les pàgines de ASP.NET estan continguts amb fitxers amb una extensió ASPX, aquests fitxers contenen etiquetes HTML⁵ o XHTML⁶. Microsoft

⁴Active Server Pages

⁵Hyper Text Markup Language

⁶eXtensible Hypertext Markup Language

recomana que per realitzar programació dinàmica es col·loqui el codi en un fitxer separat o en una etiqueta de script especialment dissenyada; els noms dels arxius estan basats amb el nom del fitxer ASPX tals com `Pagina.aspx.cs` o `Pagina.aspx.vb`. Aquest model de programació s'anomena *code-behind*. Les pàgines de ASP.NET també són conegudes com "Web forms".

- **Visual Basic .NET:** és un llenguatge de programació orientat a objectes que es pot considerar com una evolució del Visual Basic. Els programes escrits amb Visual Basic .NET, a l'igual que tots els llenguatges de programació basats amb .NET, requereixen del Framework .NET per executar-se.

La sintaxi del Visual Basic .NET i el Visual Basic és pràcticament la mateixa, però també té nous afegits, com per exemple, el control estructurat d'excepcions i la programació orientada a objectes. Les diferències entre els dos llenguatges són profundes, sobre tot en les biblioteques, però tot i així segueixen mantenint una gran similitud.

- **Microsoft Visual Studio 2005:** és un IDE⁷ que suporta diferents llenguatges de programació tals com Visual C++, Visual C#, Visual J#, ASP.NET i Visual Basic .NET. Aquest IDE, és un IDE desenvolupat per funcionar amb sistemes operatius Windows i accepta plug-ins que milloren la seva funcionalitat. Permet als desenvolupadors crear aplicacions, llocs Web, aplicacions Web i serveis Web.

Visual Studio com qualsevol IDE, incorpora un editor de codi que suporta el ressaltat de sintaxi i el auto-completat de codi amb IntelliSense. També inclou un depurador que pot funcionar tant com a depurador de nivell font i com a nivell màquina. Altres eines integrades són un dissenyador de formularis i un dissenyador de classes que s'utilitza per crear i editar classes; aquest dissenyador pot generar codi C# i VB.NET. A la figura 2.4 es pot observar una imatge de l'entorn de desenvolupament del Visual Studio.

⁷entorn de desenvolupament integrat

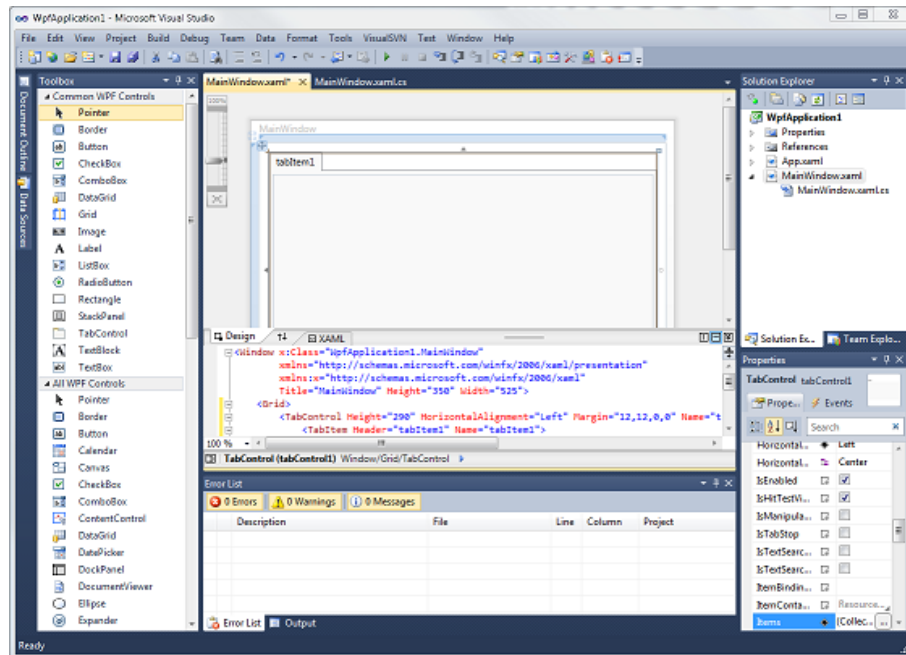


Figura 2.4: Entorn de desenvolupament del Visual Studio

- **Microsoft SQL Server Express 2005:** és per ús lliure i distribuïble als desenvolupadors de software, és la versió lleugera de Microsoft SQL server. El Microsoft SQL Server és un sistema per la gestió de base de dades desenvolupada per Microsoft, és l'alternativa a altres potents sistemes de gestió de bases de dades, com per exemple, Oracle, PostgreSQL, MySQL. Els seus llenguatges per realitzar consultes són T-SQL i ANSI SQL.
- **JavaScript:** és un llenguatge de scripting basat amb objectes, es va dissenyar amb l'idea que tingués una sintaxi similar al llenguatge de programació Java, és utilitzat per accedir a objectes en aplicacions, permeten el desenvolupament d'interfícies d'usuari millorades i pàgines Web dinàmiques.

Tots els navegadors Web interpreten el codi JavaScript integrat dins de les pàgines Web, es pot incloure en qualsevol document i és compatible amb HTML, PHP⁸ i ASP. Tradicionalment, s'utilitza amb pàgines Web HTML,

⁸Hypertext Preprocessor

per realitzar operacions sense accés a funcions del servidor.

La W3C⁹ defineix el mètode correcte per afegir codi JavaScript en una estructura HTML, ja que afegir-lo directament és una pràctica no recomanada, el mètode correcte és afegir codi JavaScript com un fitxer extern, tan per qüestions d'accessibilitat i velocitat de navegació.

- **Cascading Style Sheets (CSS):** és un llenguatge utilitzat per a definir la presentació d'un document HTML. El W3C és l'encarregat de formular l'especificació dels fulls d'estil que servirà d'estàndard per als navegadors.

L'idea del desenvolupament de CSS és separar l'estructura d'un document de la seva presentació. L'informació d'estil pot ser adjuntada tant com un document separat o en el mateix document HTML. En aquest document, poden definir-se estils generals en la capçalera del document o en cada etiqueta particular mitjançant l'atribut "style".

Alguns dels avantatges d'utilitzar CSS és tenir un control centralitzat de la presentació d'un lloc Web i que el document HTML en si mateix és més clar d'entendre i s'aconsegueix reduir considerablement la seva grandària.

2.2 Requeriments

Per tal de poder realitzar el projecte serà de vital importància disposar d'una màquina en la qual pugui funcionar el Microsoft Visual Studio 2005 Professional Edition i el Microsoft SQL Server 2005 Express Edition i que tingui instal·lat el Microsoft .NET Framework 2.0.

- Requeriments Microsoft Visual Studio 2005 Professional Edition
 - Processador: Mínim 600MHz, Recomanat 1GHz.
 - Memòria RAM: Mínim 192MB, Recomanat 256MB.
 - Espai en el disc dur: 2GB.

⁹World Wide Web Consortium

- Sistema Operatiu:
 - * Microsoft Windows 2000 Service Pack 4.
 - * Microsoft Windows XP Service Pack 2.
 - * Microsoft Windows XP Professional x64.
 - * Microsoft Windows Server 2003 Service Pack 1.
 - * Microsoft Windows Server 2003 x64.
 - * Microsoft Windows Server 2003 R2.
 - * Microsoft Windows Server 2003 R2, x64.
 - * Microsoft Windows Vista.
- Requeriments Microsoft SQL Server 2005 Express Edition
 - Processador: Mínim: 500MHz, Recomanat: 1GHz.
 - Memòria RAM: Mínim: 192MB, Recomanat: 512MB.
 - Espai en disc dur: 600MB.
 - Sistema Operatiu:
 - * Microsoft Windows Server 2003 Standard Edition.
 - * Microsoft Windows Server 2003 Enterprise Edition.
 - * Microsoft Windows Server 2003 Datacenter Edition.
 - * Microsoft Windows Small Business Server 2003 Standard Edition.
 - * Microsoft Windows Small Business Server 2003 Premium Edition.
 - * Microsoft Windows 2000 Server.
 - * Microsoft Windows 2000 Advanced Server.
 - * Microsoft Windows 2000 Datacenter Server.
 - * Microsoft Windows XP Professional.
 - * Microsoft Windows XP Media Edition.
 - * Microsoft Windows XP Tablet Edition.

- * Microsoft Windows 2000 Professional.
- * Microsoft Windows XP Home Edition.
- * Microsoft Windows Server 2003 Web Edition.

La màquina amb la que es treballa, és una màquina amb un processador de 2GHZ, amb 1GB de memòria RAM, amb més de 2GB d'espai en el disc dur, amb un Sistema Operatiu Microsoft Windows XP Professional Service Pack 3 i es troba instal·lat el software Microsoft .NET Framework v3.5.

2.3 Planificació i fases

La planificació del projecte està dividida en tres grans tasques, la primera consisteix en fer un aprenentatge de la plataforma .NET. La segona en fer el desenvolupament del framework. L'última tasca consisteix en fer una petita aplicació Web utilitzant les eines desenvolupades.

Tot seguit es fa una explicació més detallada de les diferents tasques d'aquest projecte i també es pot veure un diagrama de Gantt a la figura 2.5.

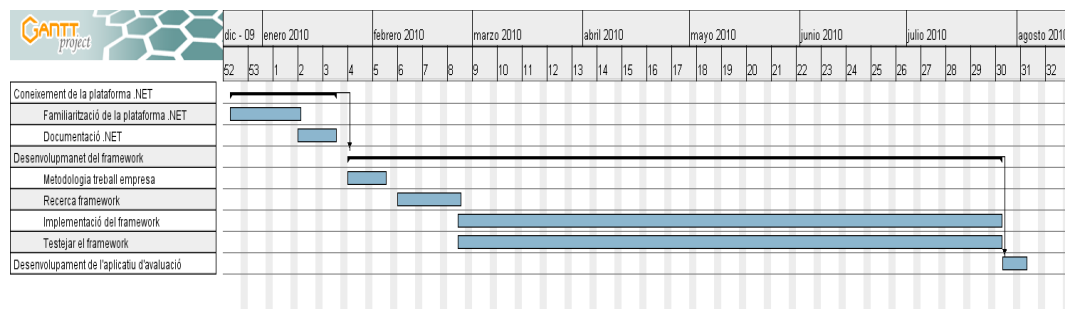


Figura 2.5: Diagrama Gantt

- **Coneixement de la plataforma .NET:** té dos subtasques ben diferenciades; una de familiarització amb la plataforma .NET i una altra de documentació .NET.
 - Familiarització de la plataforma .NET
Degut al desconeixement del funcionament de la plataforma .NET, el

primer pas és aprendre unes nocions bàsiques d'aquesta plataforma, com per exemple la sintaxi, alguns dels controls que existeixen i com utilitzar-los. Per tal de realitzar aquest aprenentatge es farà una petita aplicació Web i un petit servei Web.

- * *Aplicació Web:* és una petita eina de gestió per una escola, destinats a dos tipus d'usuaris amb perfils ben diferenciats. Hi ha un perfil d'administrador, que permet afegir, modificar i esborrar dades de contacte o dades acadèmiques dels alumnes. El perfil d'usuari, permet que un alumne pugui consultar el seu expedient acadèmic. Mirar annex A per una explicació més detallada.
- * *Servei Web:* és una eina molt bàsica i molt simple per veure com funcionen els serveis Web. Aquest servei et retorna el resultat d'una operació matemàtica que es vol realitzar a partir de dos números introduïts. Mirar annex B per una explicació més detallada.

– Documentació .NET

Per tal d'adquirir un millor coneixement de la plataforma .NET, es farà un treball de documentació dels diferents controls ASP.NET. Es poden distingir entre diferents tipus de controls.

- * *Controls estàndard de servidors Web ASP.NET:* és una gran selecció de controls bàsics que es poden utilitzar en formularis; com per exemple, botons, quadres de text, controls per afegir imatges, llistes desplegable i controls més complexes, com un calendari. Una explicació més detallada de cada un dels diferents controls es troba a l'apartat [8] de la bibliografia.
- * *Controls ASP.NET de servidor Web de dades:* són controls que poden ser enllaçats a controls d'origen de dades per facilitar l'operació de mostrar i modificar dades de l'aplicació Web. Per exemple es pot enllaçar a una taula de clients que contingui les dades personals de tots els clients d'una empresa. Existeixen diferents controls de servidor Web de dades, que tenen unes

característiques diferents. Per exemple, n'hi han que et mostren els resultats amb forma de taula i que depenent del control et permeten fer operacions d'inserció, edició, actualització o eliminació de registres i d'altres que et mostren els resultats amb forma de llista de només lectura. Per més informació de cada un dels diferents controls mirar a l'apartat [9] de la bibliografia.

- * *Controls de servidor Web d'origen de dades:* són controls que s'utilitzen per associar-se a un origen de dades per facilitar la presentació i modificació de dades en les aplicacions Web. Aquests controls permeten treballar amb diferents tipus d'origen de dades, com bases de dades, arxius XML, etc. Permeten connectar-se a un origen de dades i permeten a altres controls enllaçar-se a ells, com per exemple els controls de servidor Web de dades, sense necessitat de codi. Si es vol veure una descripció detallada de cada un dels control, mirar a l'apartat [10] de la bibliografia.
- * *Controls de validació ASP.NET:* és un conjunt de controls que proporcionen una forma fàcil i eficaç per comprovar errors en la informació introduïda pels usuaris i mostrar missatges si és necessari. Per saber com utilitzar aquests controls, consultar l'apartat [11] de la bibliografia.
- * *Controls ASP.NET d'exploració:* diferents controls que s'utilitzen per obtenir ajudes per l'exploració de pàgines Web. Existeixen tres controls diferents; el primer és el Menu, aquest admet menús i submenú i també et permet crear menús dinàmics. L'altre és el SiteMapPath, que mostra una ruta de desplaçament a l'usuari i la ruta de retorn a la pàgina principal. I per últim, tenim el TreeView, que s'utilitza per mostrar dades jeràrquiques amb un estructura d'arbre. Per obtenir més informació mirar a l'apartat [12] de la bibliografia.

* *Controls ASP.NET d'inici de sessió*: aquests controls proporcionen una forma còmoda per l'inici de sessió, la recuperació de contrasenyes i la creació de nous usuaris de les pàgines que ho requereixin. Per una descripció més detallada consultar a l'apartat [13] de la bibliografia.

- **Desenvolupament del framework**: està dividida en quatre subtasques, la primera subtasca consisteix en fer-se una idea de la metodologia de treball de l'empresa, la segona amb fer una recerca de diferents frameworks similars als que es vol desenvolupar en aquest projecte, amb l'objectiu de poder extreure algunes idees, la tercera en implementar el framework i l'última en testejar el correcte funcionament de les noves eines implementades en el framework.

- Metodologia treball empresa

Aquesta fase consisteix a mirar-se el codi font d'una aplicació Web desenvolupada per l'empresa, el seu objectiu és assolir l'esquema de treball de l'empresa i la seva metodologia mirant quins són els controls .NET més utilitzats i com els utilitzen.

- Recerca framework

Aquesta fase consisteix en fer un estudi de diferents frameworks per aplicacions Web que es poden trobar avui en dia; amb l'objectiu de fer-se una idea a l'hora d'implementar les eines necessàries del nou framework.

* *Django*: és un framework fet amb Python; entre les seves característiques destaca un patró molt similar al MVC¹⁰, llicència BSD¹¹ i ús d'un ORM¹² per l'accés a base de dades.

Els paràmetres de la connexió a una base de dades estan definits en un fitxer que s'ha de modificar algunes opcions, com per exemple, el gestor de la base de dades, el nom de la base de dades,

¹⁰Model Vista Controlador

¹¹Berkeley Software Distribution

¹²Object-relational mapping

l'usuari, la contrasenya de l'usuari i la direcció IP on es troba instal·lat el gestor de la base de dades.

La vista està formada per plantilles que contenen codi HTML. Les vistes es defineixen en un fitxer que especifiquen quina plantilla utilitzar. Els noms de les vistes han de coincidir amb els valors definits en un fitxer que mitjançant expressions regulars decideix quina vista utilitzar per pintar en pantalla cada pàgina. Una millor documentació del framework Django es troba a l'apartat [14] de la bibliografia.

Realment, aquest framework no serà de massa utilitat a l'hora de fer-se una idea de com implementar les eines del nou framework, ja que es centra en fer més àgil el desenvolupament d'una aplicació Web basada en MVC, en lloc de definir nous tipus de controls.

- * *Framework d'empresa X*: és un framework implementat per una empresa, on s'han definit nous controls per agilitzar el desenvolupament d'aplicacions Web. Han implementat nous controls per formularis, per l'accés a base de dades, per validar expressions típiques com per exemple un DNI, una data, una hora, una contrasenya, etc.

Aquest framework serà bastant útil a l'hora de fer-se una idea del nou framework a desenvolupar, per aquest motiu s'implementarà una aplicació Web utilitzant les seves llibreries; es pot veure l'aplicació implementada a la secció C dels annexes.

– Implementació del framework

Aquesta és la subtasca més creativa, més important i de més durada del projecte, ja que si recordem, el projecte consisteix en fer un framework per realitzar aplicacions Web; per tant, serà en aquesta subtasca on es desenvoluparà el framework.

En aquesta fase es tindrà que implementar les noves eines esmentades en els objectius del projecte. Les noves eines que s'implementaran,

es faran en funció als controls que acostuma a utilitzar i com els utilitzen els programadors de l'empresa en les aplicacions Web desenvolupades.

– Testejar el framework

Aquesta fase és en la que es comprovarà el correcte funcionament de les eines desenvolupades a la fase anterior. Aquesta, es durà a terme de forma simultània amb l'anterior fase, a mesura que es vagin implementant noves eines es comprovarà el seu funcionament.

- **Desenvolupament de l'aplicatiu d'avaluació:** aquesta tasca consisteix en desenvolupar una petita aplicació pràctica utilitzant les noves eines implementades en el framework i d'aquesta manera poder fer una demostració del funcionament de les noves eines i també poder fer l'avaluació del temps invertit en el desenvolupament de l'aplicació per tal de determinar si amb aquestes noves eines es millora el temps de producció.

2.4 Disseny

L'idea d'aquest projecte és implementar una llibreria DLL¹³ que contingui els diferents controls i agregar-la al projecte de la nova aplicació Web que es vol desenvolupar, per tal de tenir accés a cada un dels controls implementats. Algunes de les funcionalitats del control, corren a càrrec de funcions implementades amb JavaScript i es dona un estil als controls mitjançant CSS. Tant les funcions de JavaScript com les classes de CSS es troben en fitxers separats.

La seqüència de la figura 2.6 és la següent:

- El programador que desenvolupa una nova aplicació Web envia una nova senyal a l'aplicació Web amb l'intenció de crear un nou control.
- L'aplicació Web envia un missatge a la llibreria DLL per tal de crear un nou control.

¹³Dynamic-link library

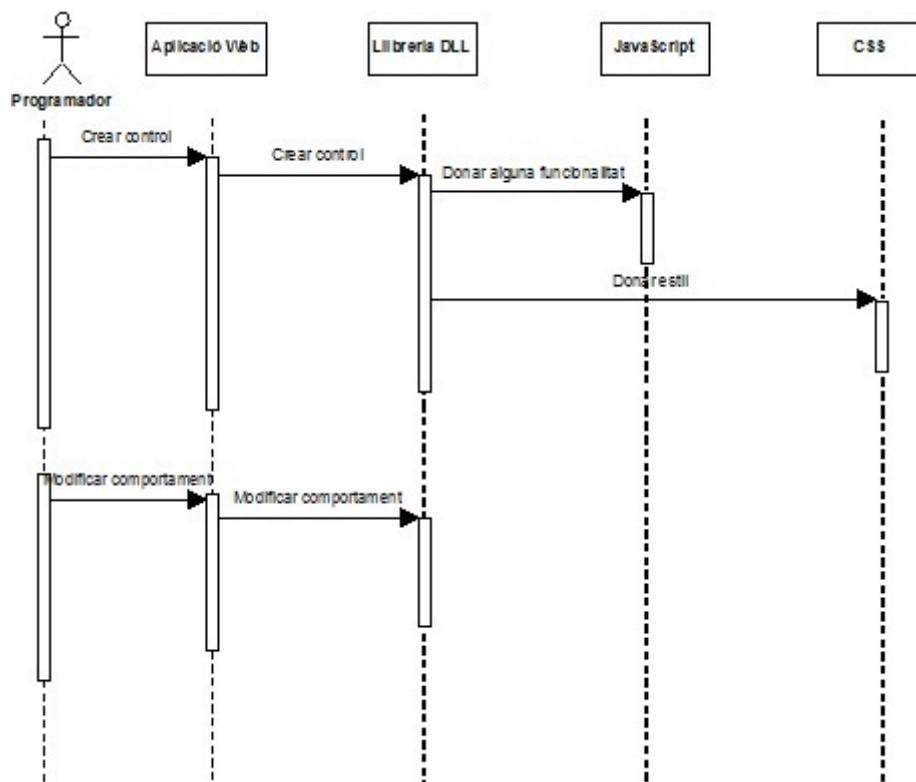


Figura 2.6: Diagrama de seqüència del framework

- La llibreria DLL crea el nou control, dóna alguna funcionalitat amb un procediment implementat amb JavaScript i li dóna un estil mitjançant les CSS.
- Si el programador vol modificar el comportament d'algun dels controls creats, li envia un missatge a l'aplicació Web.
- L'aplicació Web envia un missatge a la llibreria DLL amb l'intenció de modificar el comportament d'algun control creat.
- La llibreria DLL modifica el comportament del control.

La llibreria DLL conté dos tipus de controls diferents, un dels tipus facilita el desenvolupament d'una aplicació de gestió Web a la capa de presentació i l'altre facilita l'operació de mostrar dades dins de l'entorn de desenvolupament. És a dir,

en el framework, s'implementen diferents controls que després seran utilitzats en el desenvolupament de noves aplicacions de gestió Web.

- **Controls de la capa de presentació:** tal com es pot observar a la figura 2.7, es pot veure un diagrama de classes dels controls orientats a la capa de presentació. Cada una de les diferents classes representa un control diferent, es pot observar en el diagrama que hi ha dotze classes implementades, tot seguit es farà una explicació detallada del disseny de cada una d'aquestes.

- Classe Data

Es pot observar que aquesta classe deriva de la classe WebControl del Framework .NET, ja que aquesta proporciona tots els mètodes, propietats i esdeveniments que són comuns a tots els controls de servidor Web. Tots els atributs d'aquesta classe estan declarats com a privats, per tal de protegir el seu accés i que només puguin accedir-hi els procediments implementats en aquesta mateixa classe. Tots els mètodes menys un estan declarats com a públics, ja que es poden cridar des d'una classe externa, l'altre mètode està declarat com a privat, ja que no interessa que s'hi pugui accedir des de fora de la mateixa classe. En els procediments es pot observar que hi ha un constructor, d'altres que permeten configurar el comportament del control, i un últim, que ens permet capturar l'esdeveniment que es produeix quan es fa un clic a l'atribut `_image`.

- Classe Hora

Igual que en la classe anterior, també deriva de la classe WebControl del Framework .NET. Es pot observar que hi han sis atributs, tots ells declarats com a privats i nou procediments declarats com a públics, en els quals un és el constructor i els altres ens serveixen per configurar el comportament del control.

- Classe DocumentIdentificatiu

Aquesta classe també deriva de la classe WebControl, tots els atributs estan declarats com a privats. Conté onze procediments, dels quals deu

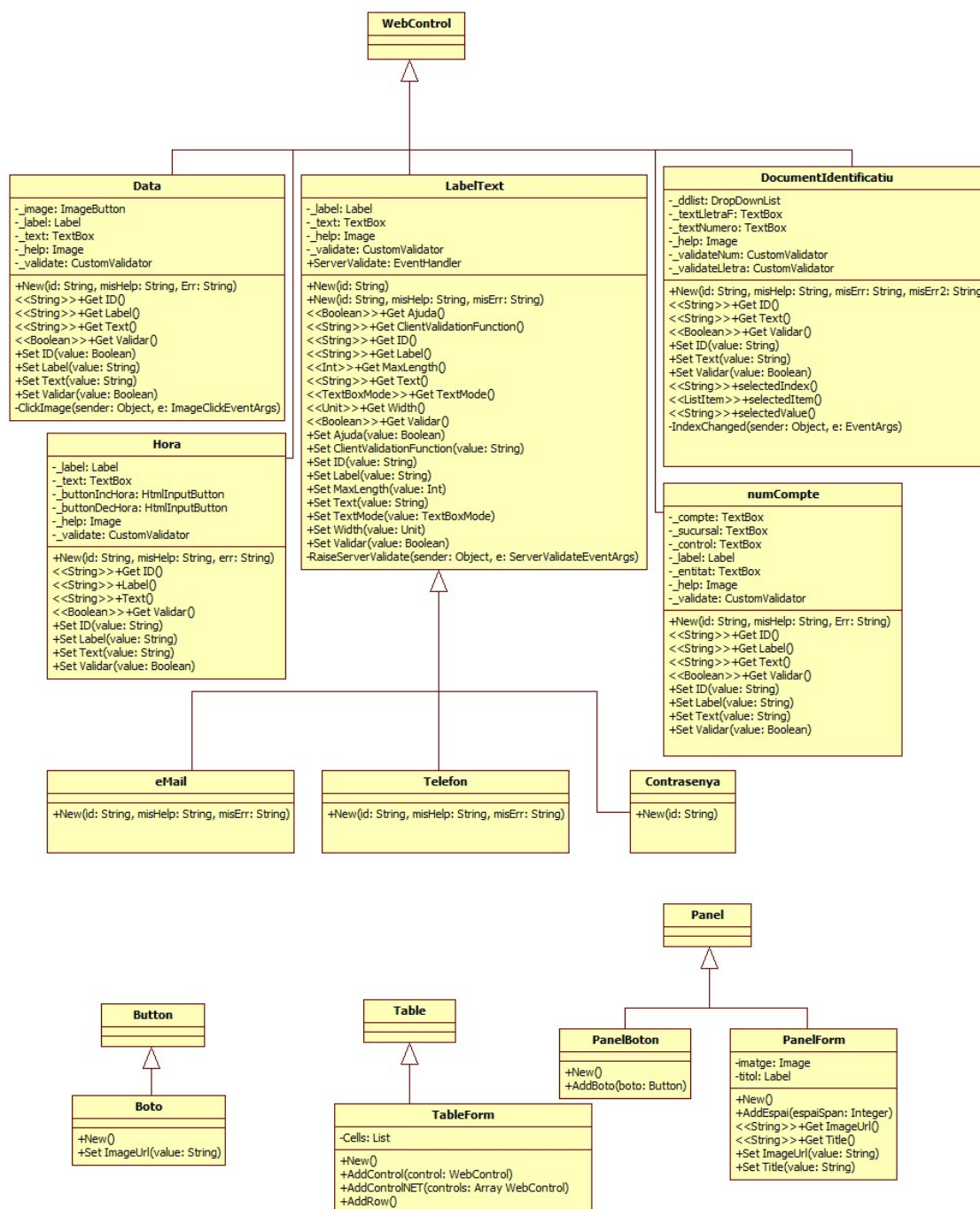


Figura 2.7: Diagrama de classes dels controls orientats a la capa de presentació

estan declarats com a públics i un com a privat, un dels procediments és un constructor. N'hi han tres que retornen l'element seleccionat de l'atribut `_ddlist` de tipus `DropDownList`. El procediment declarat com a privat és el responsable de capturar l'esdeveniment que es produeix quan es canvia l'element seleccionat de la `_ddlist` i els altres permeten configurar el comportament del nou control.

– Classe NumCompte

La següent classe conté set atributs privats i nou procediments públics. Igual que en les classes anteriors també deriva de la classe `WebControl`. Entre els procediments hi ha un constructor i la resta permeten configurar el comportament del nou control.

– Classe LabelText

Aquesta classe deriva de la classe `WebControl`, està formada per quatre atributs privats, un de públic, vint procediments públics i un de privat, dels públics, hi ha dos constructors i la resta permeten modificar el comportament i part de l'estil del control. El procediment privat és el responsable de llançar l'esdeveniment que es produeix quan es vol fer validació pel costat del servidor.

– Classe eMail, Telefon i Contrasenya

Es pot observar que aquestes classes només tenen un procediment declarat com a públic, que és el constructor. Deriven de la classe anterior `LabelText`, s'ha dissenyat així, ja que les tres classes utilitzen els procediments i atributs de la classe pare, però es comporten de manera diferent, en el constructor de cada una de les classes es configura el comportament i l'estil dels diferents controls.

– Classe Boto

En la següent classe es pot observar que deriva de la classe `Button` del Framework .NET, ja que el seus mètodes són els mateixos que la classe pare, la diferència és que s'ha afegit una nova funcionalitat i el comportament del control varia una mica, ja que se li permet posar una imatge al nou control. La classe està formada per dos

mètodes declarats com a públics, un és el constructor i l'altre és la nova funcionalitat que s'ha afegit.

– Classe TableForm

La classe TableForm deriva de la classe Table del Framework .NET, ja que els procediments són els mateixos que la classe base, amb aquesta, se li afegit alguna funcionalitat respecte a la classe base i s'ha modificat lleugerament el seu comportament. Conté quatre procediments declarats com a públics, un és el constructor i els altres les noves funcionalitats que s'han implementat, que permeten afegir controls a la taula d'una forma més senzilla i transparent per part del programador. Només té un atribut declarat com a privat.

– Classe PanelBoton

Aquesta classe deriva de la classe Panel del Framework .NET, per tal d'aprofitar els procediments de la classe pare, amb la diferència que se li afegeix una nova funcionalitat, aquesta nova funcionalitat és un mètode declarat com a públic, que permet afegir controls del tipus Button al nou control. L'altre procediment és el constructor que també està declarat com a públic.

– Classe PanelForm

Es pot observar que de forma similar a la classe anterior també deriva de la classe Panel del framework .NET. S'ha modificat el comportament i afegit una nova funcionalitat respecte la classe pare. La classe està formada per dos atributs declarats com a privats, que ens permet posar un títol i una imatge a la part superior del panell. També està formada per sis procediments, dels quals un és el constructor, els altres quatre ens permeten configurar el comportament del nou control i l'últim, és una nova funcionalitat que s'ha afegit per tal que sigui més senzill i elegant pel programador afegir un espai entre control i control.

- **Controls enllaçats a dades:** es pot veure a la figura 2.8 un diagrama de classes dels controls enllaçats a dades. Només hi ha una classe que representa l'únic control que permet mostrar les dades d'un origen de dades.

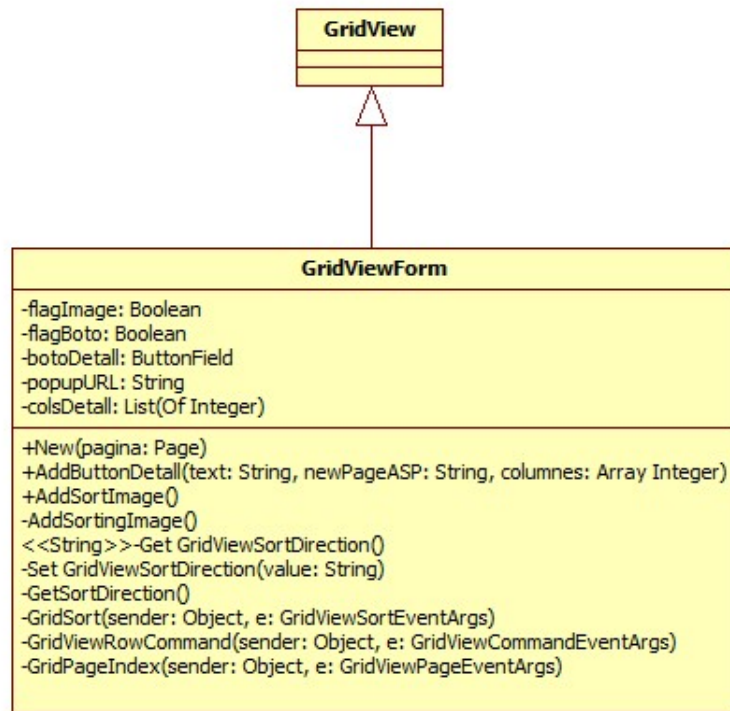


Figura 2.8: Diagrama de classes dels controls enllaçats a dades

– Classe GridViewForm

Aquesta classe deriva de la classe GridView del Framework .NET, ja que el control GridViewForm es comporta igual que el control GridView, però s'ha implementat alguna funcionalitat que per defecte el control GridView no té. Es pot observar al diagrama de classes que aquesta classe té cinc atributs declarats com a privats, ja que només es vol permetre l'accés amb aquests des dels procediments de la mateixa classe. També té deu procediments, dels quals n'hi ha tres de públics, on un és el constructor de la classe i els altres permeten modificar el

comportament del control. Els altres procediments estan declarats com a privats, perquè no es vol permetre que s'accedeixin a ells des d'una altra classe. Entre els procediments privats, n'hi ha que permeten capturar i tractar esdeveniments i d'altres que poden afegir una imatge d'ordenació a les capçaleres del GridViewForm o coneixer la direcció d'ordenació de dades.

2.5 Implementació

En aquest apartat, s'explica com s'ha implementat cada un dels mètodes de cada una de les classes. Per l'implementació s'han utilitzat els llenguatges Visual Basic .NET, JavaScript i CSS. La raó de l'elecció del VB.NET¹⁴ enlloc del C# és perquè a l'empresa treballen amb aquest llenguatge. El JavaScript s'ha utilitzat perquè és un llenguatge que s'executa al costat del client, a diferència del VB.NET que corre al costat del servidor i d'aquesta manera s'allibera el servidor de realitzar algunes tasques i l'execució és més ràpida. Les CSS s'han utilitzat per definir la presentació dels nous controls.

2.5.1 Visual Basic .NET

Amb aquest llenguatge s'ha implementat la llibreria DLL, que està formada per cada una de les classes respectives a cada un dels diferents controls.

- **Controls de la capa de presentació**

A continuació es pot veure una detallada explicació de com s'ha implementat cada un dels procediments de les classes pertanyents als controls destinats a ser utilitzats en la capa de presentació.

- Classe Data: aquesta classe representa un control que permet a l'usuari introduir una data, es pot introduir manualment o pitjant un boto que et desplega un calendari, a on l'usuari pot seleccionar la data desitjada, un cop s'ha introduït, es comprova que les dades introduïdes siguin

¹⁴Visual Basic .NET

correctes i sinó ho són, es mostra un missatge d'error. També conté una imatge que quan es passa el punter del ratolí per damunt mostra una ajuda a l'usuari. La classe està formada per cinc atributs, un és un control Label i ens mostra un text per pantalla, l'altre és un TextBox i permet a l'usuari introduir les dades, un altre atribut, és un control ImageButton i és el boto que ha de desplegar el calendari, un altre és un Image i és el responsable de mostrar l'ajuda i l'últim, és un control CustomValidator i és l'encarregat de validar les dades introduïdes per l'usuari.

- * *New(id,misHelp,Err)*: al constructor se li passa tres arguments per valor, el primer és l'identificador. Els altres dos, són els missatges d'ajuda i d'error. Tal com es pot veure a la figura 2.9, el que fa aquest procediment és cridar als constructors de cada un dels diferents atributs i configurar el seu el comportament i part de l'estil. Alguna de les funcionalitats, com per exemple, desplegar el calendari, ensenyar o eliminar el missatge d'ajuda i comprovar que les dades introduïdes per l'usuari siguin correctes va a càrrec de procediments implementats amb JavaScript. També s'associa un esdeveniment amb un controlador d'esdeveniments en temps d'execució mitjançant l'instrucció AddHandler. Finalment s'afegeix els diferents controls creats, a la classe.
- * *ID()*: és un procediment identificat amb la paraula clau Property, que inclou un mètode get i un altre set. El mètode get el que fa és retornar l'identificador i el set assigna un identificador. El procediment ID() està declarat amb l'instrucció Overrides ja que reemplaça al de la classe base.
- * *Label()*: de la mateixa manera que el procediment anterior, inclou un mètode set i un de get, aquests permeten assignar i retornar el text del control Label. Per tal d'assignar i retornar un text es crida la propietat Text de la classe Label.

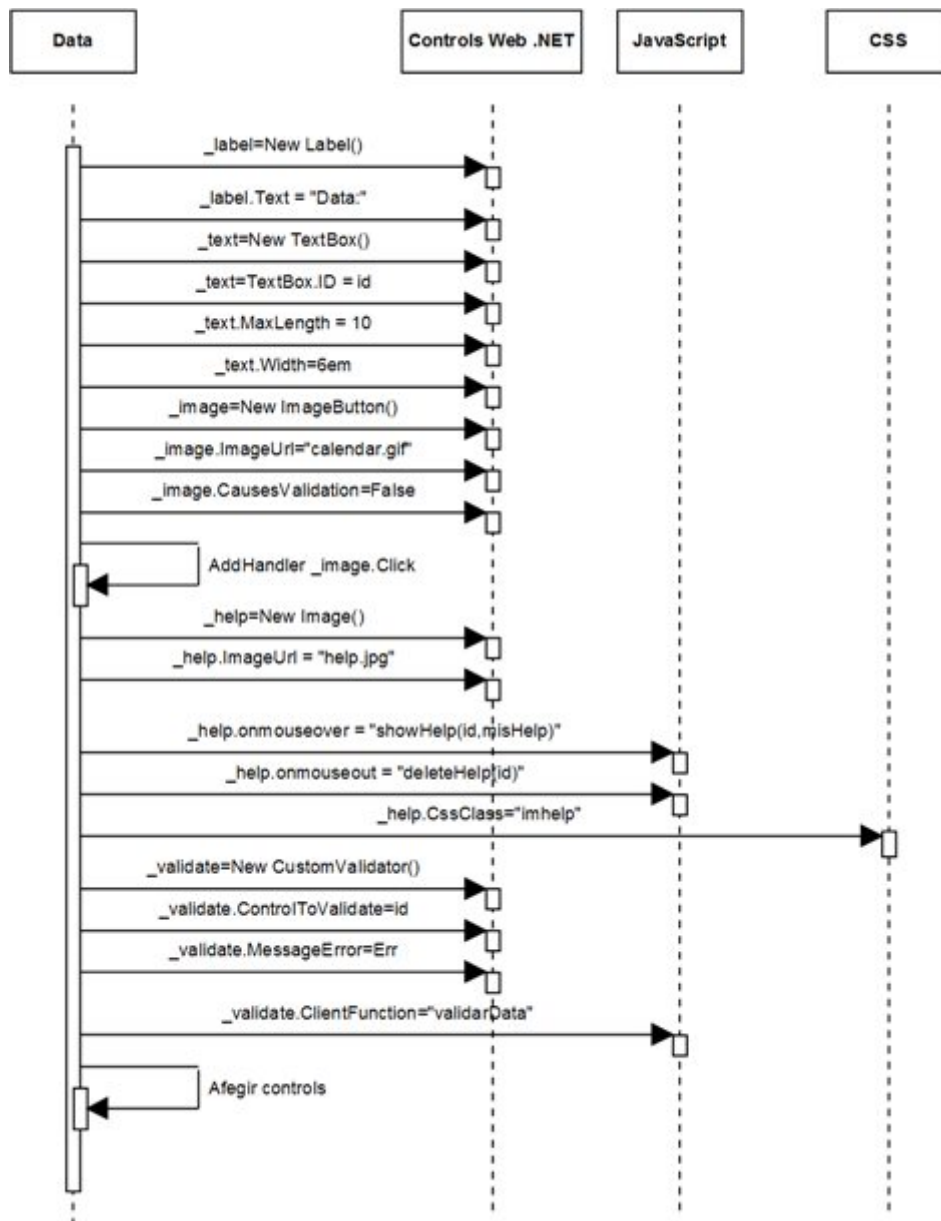


Figura 2.9: Diagrama de seqüència del constructor de la classe Data

- * *Text()*: propietat que inclou un mètode get i un de set, que retorna o assigna un text al control TextBox. Per tal de que es pugui assignar o retornar un text es fa ús de la propietat Text de la classe TextBox.

- * *Validar()*: aquesta propietat permet retornar si el control valida les dades introduïdes per l'usuari o activar i desactivar la validació del control. Com totes les propietats també conté dos mètodes, el mètode *get*, que és el responsable de retornar l'estat d'activació de la validació del control, retornant la propietat *Enabled* de la classe *CustomValidator*. L'altre mètode, és el *set*, que permet activar o desactivar la validació del control assignant *True* o *False* a la propietat *Enabled* de la classe *CustomValidator*.
 - * *ClickImage(sender,e)*: aquest procediment és el responsable de capturar l'esdeveniment generat al clicar l'atribut de tipus *ImageButton*. El que fa el procediment és cridar el mètode *scwShow* d'un fitxer extern de JavaScript. Per tal de cridar una funció JavaScript s'utilitza el procediment *RegisterStartupScript* de la classe *ScriptManager*.
- Classe Hora: representa un control que permet a l'usuari introduir una hora, es pot introduir manualment o clicant uns botons que incrementen i decrementen les hores o els minuts depenent de la posició del cursor. Si el format de l'hora introduïda és incorrecta et mostra un missatge d'error. També té una imatge d'ajuda que quan l'usuari passa el punter del ratolí per damunt mostra l'ajuda del nou control. La classe està formada per sis atributs, cada un d'ells és un control, aquests són, un control *Label*, que mostra el text per pantalla, un *TextBox*, que permet a l'usuari introduir les dades, dos controls *HtmlInputButton*, que permeten a l'usuari incrementar i decrementar l'hora, un *Image*, que és el responsable de mostrar l'ajuda i un *CustomValidator*, que és el que valida les dades introduïdes per l'usuari. Els botons d'incrementar i decrementar l'hora són controls *HtmlInputButton*, ja que així s'evita el *PostBack*.
- * *New(id,misHelp,Err)*: de la mateixa forma que el constructor de la classe anterior, també se li passa tres argument per valor, on el primer és l'identificador, el segon el missatge d'ajuda i el

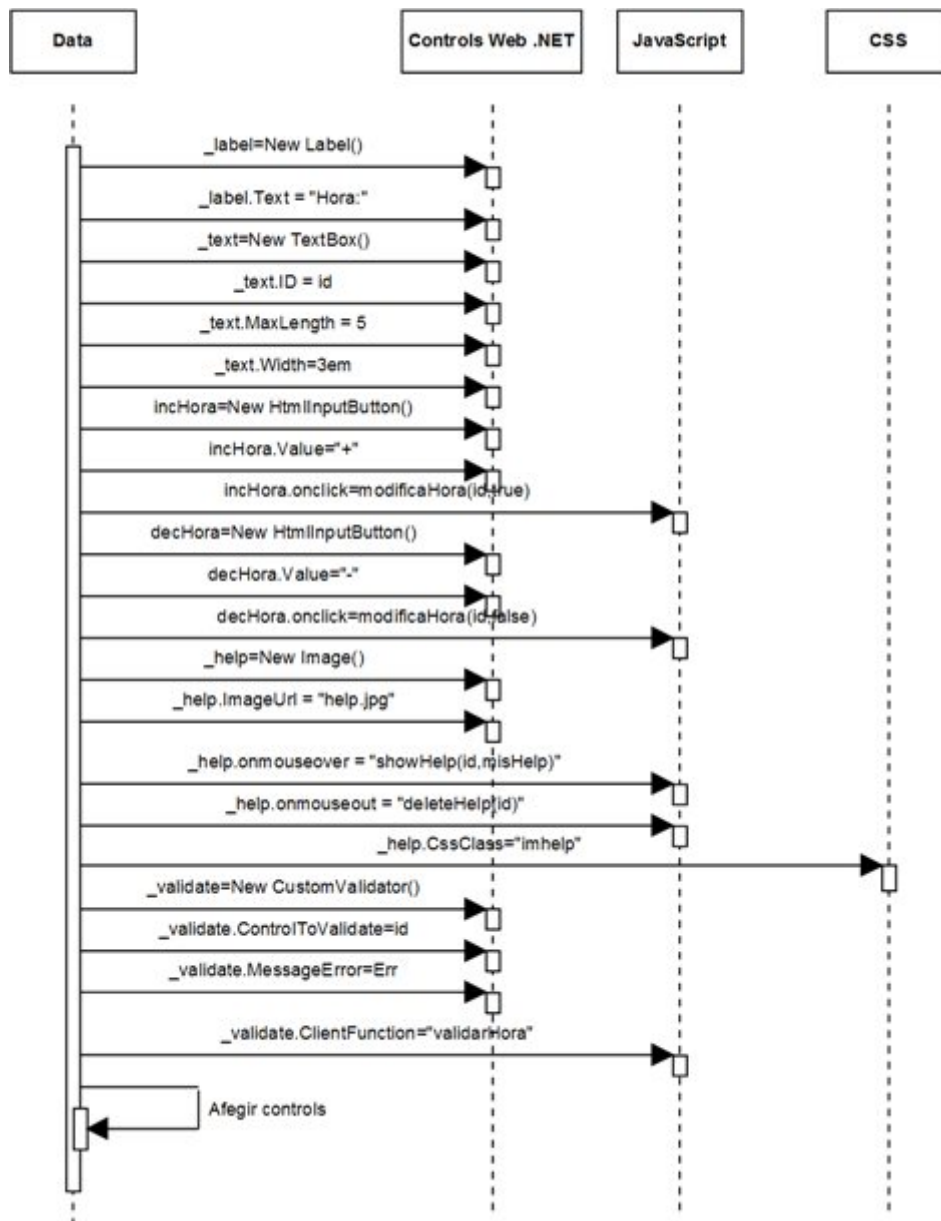


Figura 2.10: Diagrama de seqüència del constructor de la classe Hora

tercer el missatge d'error. A la figura 2.10 es pot observar que el constructor d'aquesta classe crida als constructors dels diferents controls i va configurant el comportament i part de l'estil de cada un d'ells. Les funcionalitats d'incrementar o decrementar l'hora,

mostrar l'ajuda i validar que les dades introduïdes siguin correctes corren a càrrec de procediments implementats amb JavaScript. Al final del constructor s'afegeix els nous controls a la nova classe.

- * *ID()*: propietat que inclou el mètode get i set. El mètode get retorna l'identificador i el set assigna un identificador. Igual que a la mateixa propietat de la classe anterior, també està declarat com a Overrides.
 - * *Label()*: propietat que mitjançant el mètode get retorna el text del control Label i mitjançant el set, assigna un text al mateix control. Per tal d'assignar i retornar el text al control Label, s'utilitza la propietat Text de la classe Label.
 - * *Text()*: aquesta propietat retorna i assigna el text del control TextBox fent ús de la propietat Text de la classe TextBox.
 - * *Validar()*: permet activar o desactivar la validació de les dades introduïdes per l'usuari, posant la propietat Enabled de la classe CustomValidator a True o False. A part de poder activar i desactivar la validació, també permet retornar l'estat d'activació de validació.
- Classe DocumentIdentificatiu: aquesta classe representa un control que està format per una llista desplegable, on l'usuari pot seleccionar quin tipus de document identificatiu vol introduir i una àrea de text perquè es pugui introduir el document identificatiu desitjat. Els tipus de documents que es poden introduir són, el NIF, el NIE i el Passaport. Igual que en les dos classes anteriors, aquesta, també valida les dades i mostra una ajuda a l'usuari. La classe està formada per sis atributs, cada un d'ells és un control, aquests són, un control Label, que mostra el text per pantalla, dos controls TextBox, que permeten a l'usuari introduir el número del document identificatiu desitjat i la lletra corresponent, un altre de tipus Image, que és el responsable de mostrar l'ajuda i els dos últims, són CustomValidator, aquests, validen el número i la lletra del document identificatiu introduït per l'usuari.

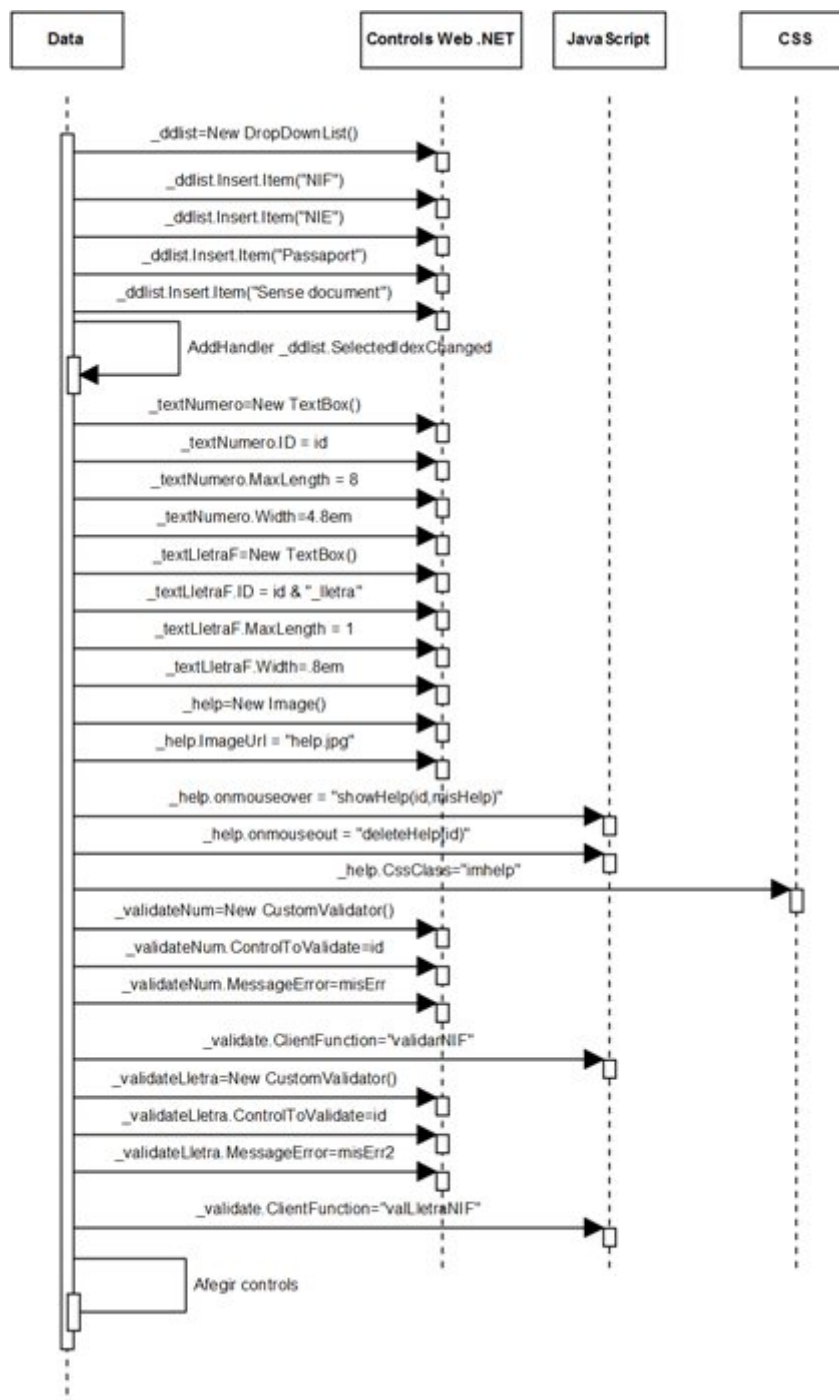


Figura 2.11: Diagrama de seqüència del constructor de la classe Document Identificatiu

- * *New(id,misHelp,misErr,misErr2)*: en el constructor d'aquesta classe, se li passa quatre arguments per valor, l'identificador, el missatge d'ajuda, el missatge d'error del número introduït per l'usuari i el missatge d'error de la lletra introduïda per l'usuari. Les funcionalitats de mostrar l'ajuda i de validar el número i la lletra, corren a càrrec de procediments JavaScript. Tal com s'observa en la figura 2.11, el constructor de la classe crida als constructors dels diferents controls i configura el comportament i part de l'estil dels controls. També s'associa l'esdeveniment que es produeix quan es canvia l'ítem seleccionat de la llista desplegable amb un controlador d'esdeveniments en temps d'execució mitjançant l'instrucció *AddHandler*. Finalment s'afegeix els diferents controls creats, a la classe.
- * *ID()*: propietat declarada amb l'instrucció *Overrides*. Aquesta propietat permet retornar i assignar un identificador mitjançant la propietat *ID* de la classe *TextBox*.
- * *Text()*: propietat que conté el procediment *get* i *set*. El procediment *get* retorna el text introduït als controls *TextBox*. El *set* assigna un valor a l'atribut *_textNumero*, que és un control *TextBox*. Per tal de retornar i assignar un valor, fa ús de la propietat *Text* de la classe *TextBox*.
- * *Validar()*: propietat que permet retornar l'estat d'activació de la validació i activar o desactivar la validació dels dos controls *CustomValidator*, mitjançant la propietat *Enabled* de la classe *CustomValidator*.
- * *IndexChanged(sender,e)*: procediment que captura l'esdeveniment produït quan es canvia l'ítem seleccionat de la llista desplegable. El que fa aquest procediment és assignar la propietat *Visible* dels controls *TextBox* i *Image* a *True* o *False*, configurar l'amplada i el número màxim de caràcters dels controls *TextBox* i assignar als *CustomValidator*, el procediment JavaScript respon-

- sable de validar les dades introduïdes per l'usuari en funció de l'ítem seleccionat de la llista desplegable.
- * *selectedItem()*: procediment que retorna l'ítem seleccionat mitjançant el mètode *selectedItem* de la classe *DropDownList*.
 - * *selectedIndex()*: procediment que retorna l'índex seleccionat mitjançant el mètode *selectedIndex* de la classe *DropDownList*.
 - * *selectedValue()*: procediment que retorna el valor seleccionat mitjançant el mètode *selectedValue* de la classe *DropDownList*.
- Classe *LabelText*: aquesta classe representa un control on es mostra un text per pantalla i permet que l'usuari introdueixi unes dades per teclat mitjançant un atribut de tipus *TextBox*, també mostra una ajuda per pantalla i permet validar les dades introduïdes per l'usuari. Aquesta classe conté cinc atributs o controls, un control *Label*, que mostra un missatge per pantalla, un *TextBox*, que permet que l'usuari introdueixi dades per teclat, una *Image*, que és el responsable de mostrar el missatge d'ajuda, un *CustomValidator*, que és l'encarregat de validar les dades introduïdes per l'usuari i un *EventHandler*, que és el responsable de controlar l'esdeveniment que permet validar del costat del servidor.
- * *New(id)*: en aquest constructor només se li passa un paràmetre per valor, l'identificador. Tal com es pot observar a la figura 2.12, aquest crida als constructors dels controls *Label* i *TextBox* i els afegeix a la classe.
 - * *New(id,misHelp,Err)*: en aquest constructor se li passa tres paràmetres per valor, l'identificador, el missatge d'ajuda i el missatge d'error de la validació de les dades introduïdes per l'usuari. Tal com s'observa a la figura 2.13, aquest crida als constructors de tots els controls i configura el comportament dels controls *Image* i *CustomValidator*. La funcionalitat d'ensenyar l'ajuda per pantalla corre a càrrec d'un procediment JavaScript. S'associa l'esdeveniment produït quan es vol validar del costat

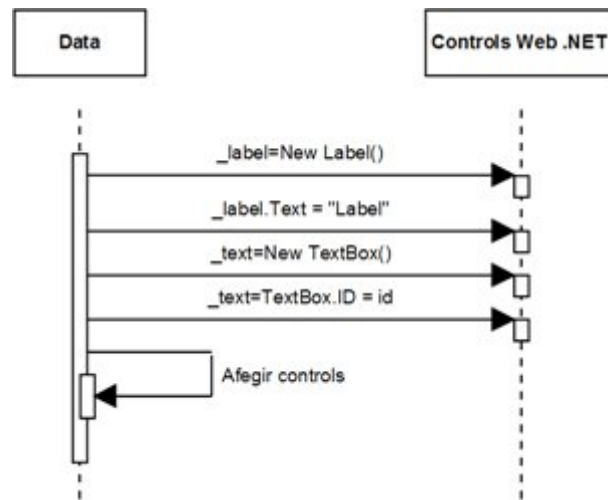


Figura 2.12: Diagrama de seqüència de primer constructor de la classe `LabelText`

del servidor, a una llista d'esdeveniments en temps d'execució, utilitzant l'instrucció `AddHandler`.

- * *ID()*: propietat declarada com a `Overrides`, ja que reemplaça al de la classe base. Aquest procediment conté el mètode `get`, que és el responsable de retornar un identificador i el `set`, que és el responsable d'assignar un identificador.
- * *Label()*: propietat que mitjançant el mètode `get` retorna el text del control `Label` i el `set` assigna un text al mateix control. Per tal de retornar i assignar un text, fa ús de la propietat `Text` de la classe `Label`.
- * *Text()*: propietat que retorna i assigna un text al control `TextBox` mitjançant la propietat `Text` de la classe `TextBox`.
- * *Width()*: procediment que conté dos mètodes, el mètode `get` i `set`. El `get` retorna l'amplada del control `TextBox` i el `set` assigna una amplada al mateix control mitjançant el procediment `Width` de la classe `TextBox`.
- * *MaxLength()*: propietat que permet retornar i assignar el número màxim de caràcters acceptats pel control `TextBox` mitjançant la

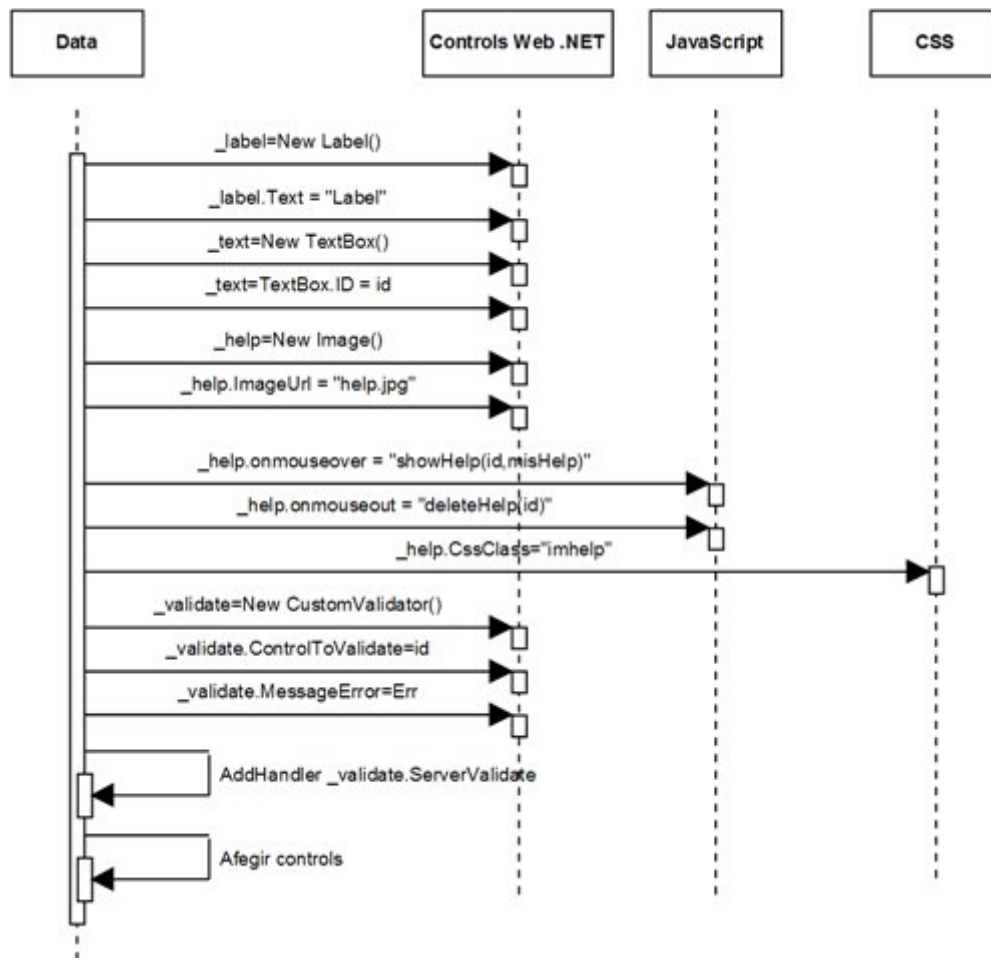


Figura 2.13: Diagrama de seqüència del segon constructor de la classe LabelText

propietat MaxLength de la classe TextBox.

- * *TextMode()*: propietat que retorna o assigna el tipus de comportament (una sola línia, multi-línia o contrasenya) del control TextBox.
- * *ClientValidationFunction()*: propietat que permet establir quina serà la funció que valida les dades introduïdes per l'usuari al costat del client. Conté dos mètodes, el get i el set, el get retorna la funció responsable de validar al costat del client i l'altre permet assignar quina és la funció que valida les dades. Per tal de poder

retornar i assignar la funció de validació, s'utilitza la propietat `ClientValidationFunction` del control `CustomValidator`.

- * *Validar()*: propietat que retorna l'estat de validació i activa o desactiva la validació de les dades introduïdes per l'usuari, mitjançant la propietat `Enabled` del control `CustomValidator`.
 - * *Ajuda()*: propietat que retorna l'estat d'ajuda del control i activa o desactiva l'ajuda mitjançant la propietat `Visible` de la classe `Image`.
 - * *RaiseServerValidate(sender,e)*: procediment responsable de capturar l'esdeveniment de validació al costat del servidor i desencadenar l'esdeveniment `ServerValidate` declarat dins de la mateixa classe.
- Classe eMail: deriva de la classe `LabelText`. Aquesta, representa un control format per diferents atributs, un d'ells, és un control `Label`, l'altre és un `TextBox`, també té un atribut que és un control `Image`, aquest, és el responsable de mostrar un missatge d'ajuda per pantalla i l'últim, és un `CustomValidator` i és el responsable de validar les dades introduïdes per l'usuari.
- * *New(id,misHelp,misErr)*: constructor de la classe `eMail`, se li passa tres arguments per valor, un és l'identificador, l'altre el missatge d'ajuda i l'últim el missatge d'error. El que fa, és cridar al constructor de la classe pare, assignar un text al control `Label` i la funció de validació, `validarMail`, al control `CustomValidator` de la classe pare. El procediment `validarMail` és un procediment implementat amb JavaScript.
- Classe Telefon: aquesta classe, igual que l'anterior, també deriva de la classe `LabelText` i està format per un control `Label`, responsable de mostrar un missatge per pantalla, un altre de tipus `TextBox`, responsable de capturar les dades introduïdes per un usuari, un control `Image`, responsable de mostrar l'ajuda i un de tipus `CustomValidator`, responsable de validar les dades. El que diferencia aquest de l'anterior,

és el número màxim de caràcters que l'usuari pot introduir, l'amplada del control TextBox, el text que es mostra per pantalla i la funció de validació.

- * *New(id,misHelp,misErr)*: constructor de la classe Telefon, igual que en el constructor de la classe anterior, també se li passa el mateixos tres arguments per valor. El que fa, és cridar el constructor de la classe pare, assignar una amplada i un número màxim de caràcters al control TextBox, assignar un text a l'atribut Label i assignar el procediment validarTel, implementat amb JavaScript, com la funció de validació del control CustomValidator.
- Classe Contrasenya: aquesta, també deriva de la classe LabelText, està formada per dos atributs, que són un control Label i un TextBox. El que diferencia aquesta classe de les dos anteriors, és que no té cap imatge per tal de mostrar l'ajuda a un usuari, tampoc valida les dades introduïdes per l'usuari i el tipus de comportament del control TextBox és un comportament com a contrasenya.
 - * *New(id)*: és el constructor de la classe Contrasenya, només se li passa un argument per valor, que és l'identificador, no se li passa cap missatge d'ajuda ni d'error, ja que aquest control no té ajuda i tampoc valida les dades introduïdes per l'usuari. El que fa, és cridar el constructor de la classe pare, assignar el text Contrasenya al control Label i assignar el comportament del l'atribut de tipus TextBox com a contrasenya o password.
- Classe numCompte: aquesta classe representa un control que permet a un usuari introduir el número de compte, validar les dades introduïdes i mostrar una ajuda. Està format per set atributs, cada un d'ells és un control, aquests són, un de tipus Label, que mostra un text per pantalla, quatre controls TextBox, que permeten a l'usuari introduir dades per teclat, cada un dels TextBox representen, l'entitat, la sucursal, el dígit de control i el número de compte. També hi ha un control Image, que

és el que mostra l'ajuda per pantalla i un CustomValidator que és el responsable de validar les dades introduïdes per l'usuari.

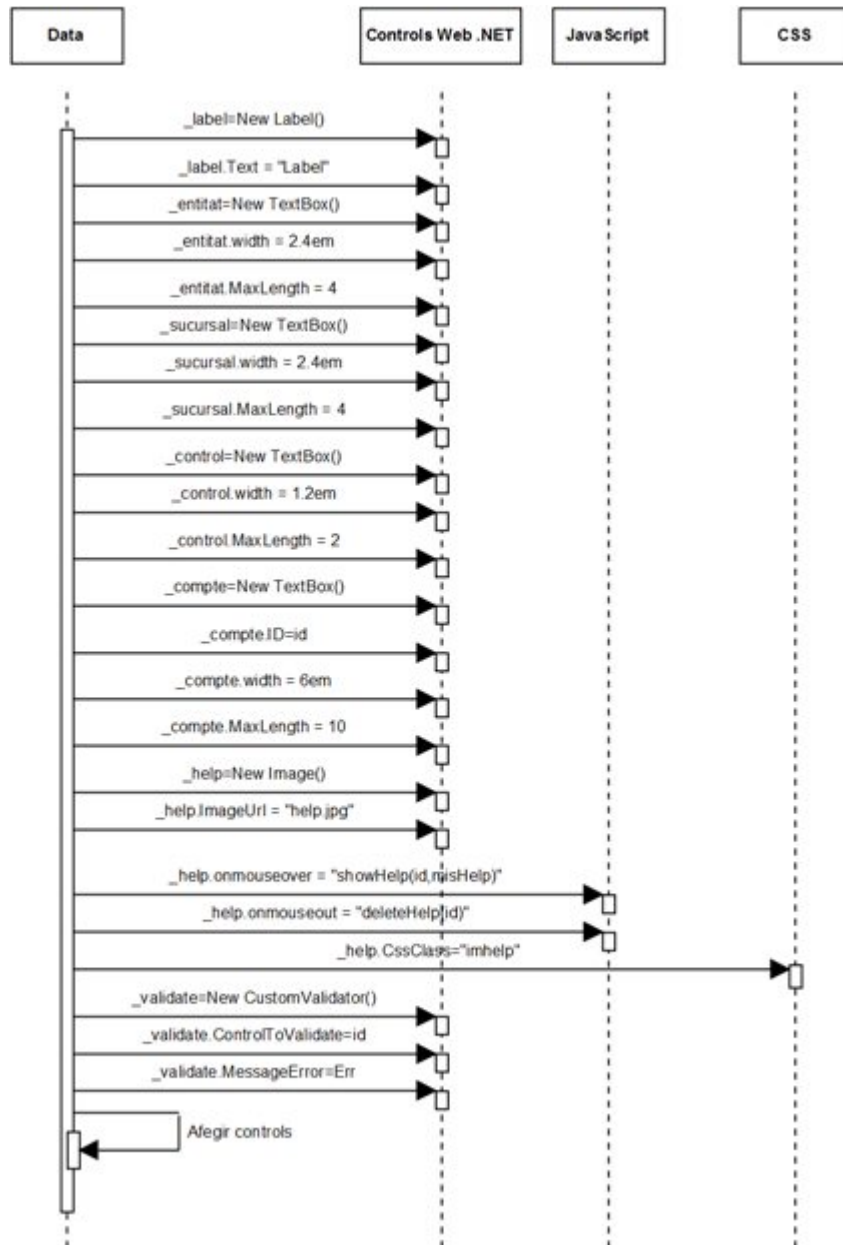


Figura 2.14: Diagrama de seqüència del constructor de la classe numCompte

* *New(id,misHelp,Err)*: se li passa tres arguments per valor, l'identificador, el missatge d'ajuda i el missatge d'error. Tal

com s'observa a la figura 2.14, aquest procediment crida a cada un dels constructors dels diferents controls, configura el seu comportament, part de l'estil i afegeix els controls a la classe. La funcionalitat de validar i mostrar el missatge d'ajuda corren a càrrec de mètodes implementats amb JavaScript.

- * *ID()*: propietat que conté dos mètodes, el *get*, que retorna l'identificador i el *set* que assigna un identificador.
 - * *Label()*: propietat que permet retornar i assignar el text del control *Label* fent ús de la propietat *Text* de la classe *Label*.
 - * *Text()*: propietat que està formada per dos procediments, el *get* i *set*. El procediment *get* retorna una concatenació de les dades introduïdes en els quatre controls *TextBox*. El mètode *set* permet assignar un text al control *TextBox* anomenat *_compte*. Com en les propietats *Text* de les classes anteriors, aquesta també fa ús de la propietat *Text* de la classe *TextBox*.
 - * *Validar()*: igual que totes les propietats anteriors, aquesta també està formada per un mètode *get* i un *set*. El *get* retorna l'estat d'activació de validació i l'altre procediment permet assignar una variable booleana a la propietat *Enabled* del control *CustomValidator*.
- Classe *TableForm*: classe que representa un control taula, permet construir una taula HTML i especificar les seves característiques d'una manera senzilla. Deriva de la classe *Table* del Framework .NET, ja que s'ha afegit unes funcionalitats noves respecte a la classe base. Les noves funcionalitats afegides són, uns procediments que permeten afegir controls a la taula d'una forma més senzilla i transparent per el programador. Té un atribut privat que és una llista de cel·les, on es guardaren les cel·les que es volen afegir a una fila de la taula.
- * *New()*: aquest constructor crida al constructor de la classe base, se li dona l'estil CSS *webtable* i es configura alguns aspectes de l'estil, com per exemple, l'amplada de la taula i el borde de la

taula.

- * *AddControl(control)*: procediment que se li passa un paràmetre per valor del tipus WebControl. És el responsable d'afegir un control del framework a la llista de cel·les mitjançant el procediment Add de la classe List(Of T).
 - * *AddControlNET(controls)*: procediment que se li passa per valor un array de controls del Framework .NET i els afegeix a la llista de cel·les.
 - * *AddRow()*: procediment que afegeix totes les cel·les de la llista de cel·les a una fila i després borra tots els elements de la llista que acaba d'afegir.
- Classe PanelForm: aquesta classe representa un control que s'utilitza per agrupar una col·lecció de controls. Aquesta deriva de la classe Panel del Framework .NET, ja que es comporta igual que la classe pare, però té alguna funcionalitat més. Les noves funcionalitats d'aquesta classe permeten afegir un títol i una imatge al panell, i a més a més, té un mètode implementat que permet afegir salts de línia entre dos controls de l'agrupació de controls. Té dos atributs, un és un control Image, que ensenya una imatge i l'altre un control Label que mostra el títol del panell.
- * *New()*: tal com s'observa a la figura 2.15, aquest procediment crida al constructor de la classe pare i li associa un estil al cos del panell, després crida als constructors dels controls Image i Label i els hi dóna un estil. Tot seguit cree una variable local, que és un panell, on se li afegeix els altres dos controls, gràcies a aquest panell, se li pot donar un estil diferent a la part on es troba el títol. Finalment afegeix aquest últim panell, al nou control.
 - * *Title()*: propietat que retorna i assigna el text del títol del panell, mitjançant la propietat Text de la classe Label.
 - * *ImageUrl()*: propietat que retorna i assigna el directori d'una imatge al panell, fent ús de la propietat ImageUrl de la classe

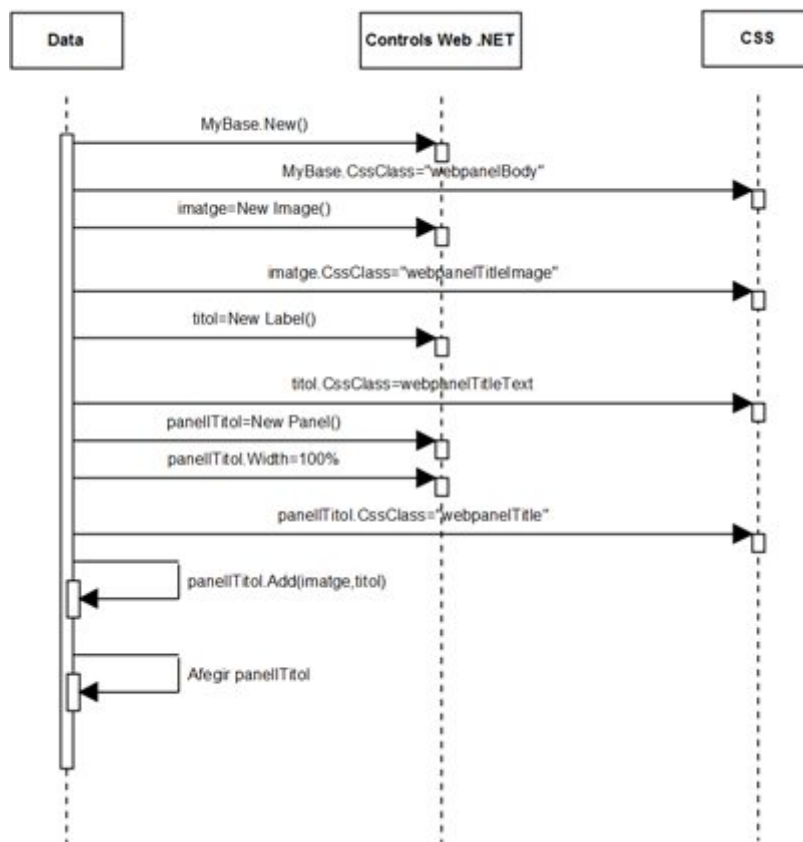


Figura 2.15: Diagrama de seqüència del constructor de la classe PanelForm

Image.

- * *AddEspai(espaiSpan)*: procediment que afegeix tants salts de línia com indica l'argument que es passa per valor, entre dos controls de l'agrupació de controls, mitjançant el control LiteralControl.
- Classe Boto: aquesta deriva de la classe Button del Framework .NET, ja que es comporta igual que la classe base, però permet afegir una imatge al boto.
 - * *New()*: constructor de la classe boto, el que fa, és cridar al constructor de la classe pare, indicar que no validi quan es faci clic al boto i associar l'estil CSS boton.

- * *ImageUrl()*: propietat solament d'escriptura definida per l'instrucció `WriteOnly`, només hi ha implementat el mètode `set`. Aquest, permet afegir una imatge al boto, afegint l'atribut arbitrari `background` a la col·lecció d'atributs arbitraris del control.
- Classe `PanelBoton`: representa un control que és una barra de botons. Aquesta deriva de la classe `Button` del Framework .NET, perquè es comporta igual, però se li afegeix l'opció d'afegir controls del tipus `Button`.
 - * *New()*: constructor de la classe, l'únic que fa, és cridar el constructor de la classe pare i associar l'estil CSS barrabotons.
 - * *AddBoto(boto)*: procediment que permet afegir controls del tipus `Button` al nou control mitjançant el procediment `Add` de la col·lecció de controls secundaris.

- **Controls enllaçats a dades**

A continuació s'explica cada un dels procediments de la classe del control `GridViewForm`, que és un control enllaçat a dades.

- Classe `GridViewForm`: representa un control que mostra les dades d'un origen de dades, a part de mostrar dades, permet ordenar-les amb una direcció ascendent o descendent. També permet fer una paginació i afegir un boto que al clicar sobre ell obre un Pop-Up¹⁵, per si es vol mostrar més informació d'una de les files de dades que mostra per pantalla. Aquesta classe, està formada per cinc atributs, dos de tipus `Boolean`, que són els responsables d'indicar si s'ha afegit el boto que obre el Pop-Up i l'imatge que indica la direcció d'ordenació, a les capçaleres de les columnes del control. El tercer atribut és el boto que obre el Pop-Up. L'altre és la direcció de la nova pàgina del Pop-Up, aquest és de tipus `String` i l'últim, és una llista de tipus `Integer`, on és guarden les columnes que es volen passar al Pop-Up.

¹⁵finestra emergent

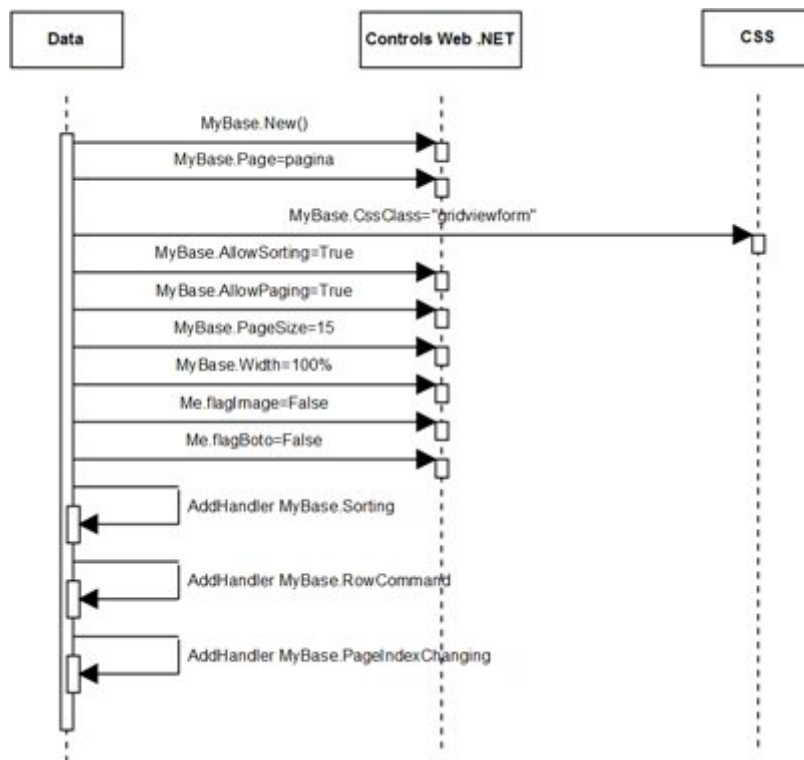


Figura 2.16: Diagrama de seqüència del constructor de la classe GridViewForm

- * *New(pagina)*: Al constructor se li passa un argument per valor, aquest, és la pàgina on s'ha afegit el control GridViewForm, se li passa, perquè és necessari per poder paginar les dades. A la figura 2.16 es pot observar que el procediment crida al constructor de la classe pare, configura el seu comportament, li associa un estil i finalment associa els esdeveniments que es produeixen a un controlador d'esdeveniments. Els esdeveniments, es produeixen quan es clica un boto de la paginació o l'hipervincle per ordenar una columna o algun dels botons que es troba al nou control.
- * *AddButtonDetall(text,newPageASP,columnnes)*: aquest procediment és el responsable d'afegir a cada una de les files de les dades que es mostren per pantalla, un boto que obra un Pop-Up. Se li passa tres arguments per valor, el primer, és el text del boto, el segon, la direcció de la pàgina del Pop-Up que s'obra i el tercer,

un array de les columnes que es passen a la pàgina del Pop-Up. Aquest procediment crida al constructor de la classe `ButtonField`, per crear el boto responsable d'obrir el Pop-Up, desactiva la propietat que indica si es realitza una validació quan es clica el boto, assigna l'argument `text` com a text del boto, l'argument `newPageASP` a l'atribut `popupURL` i l'atribut `flagBoto` a `True`. També afegeix el boto a cada una de les files mitjançant el procediment `Add` de la col·lecció de columnes del control `GridView` i per finalitzar, afegeix cada una de les columnes passades per valor com un argument, a una llista d'enters.

- * *`AddSortImage()`*: procediment responsable d'assignar el valor `True` a l'atribut `flagImage` i cridar el procediment `AddSortingImage` perquè afegeixi l'imatge d'ordenació a les capçaleres de les columnes del nou control.
- * *`AddSortingImage()`*: aquest procediment afegeix una imatge d'ordenació a les capçaleres de les columnes del control `GridViewForm`. El procediment comprova si s'ha afegit el boto que obra un Pop-Up o els botons d'editar, d'eliminar i seleccionar, que proporciona el mateix control `GridView`, depenent de si s'ha afegit algun d'aquests botons es comença a afegir l'imatge per una columna o un altra. Abans d'afegir l'imatge d'ordenació amb alguna de les columnes, també comprova la direcció d'ordenació i depenent d'ella, afegeix un tipus d'imatge.
- * *`GridViewSortDirection()`*: propietat formada per dos mètodes, el `get` i `set`. El mètode `get` retorna la direcció d'ordenació, amb el cas que no s'hagi assignat cap direcció d'ordenació retorna que la direcció és ascendent. El `set` assigna una direcció ascendent o descendent com la direcció d'ordenació.
- * *`GetSortDirection()`*: funció que canvia la direcció d'ordenació i retorna la nova direcció.

- * *GridSort(sender,e)*: procediment responsable de capturar l'esdeveniment que es produeix quan es clica un hipervincle d'ordenació i ordenar el DataView que representa una vista personalitzada de la taula de dades de memòria DataTable. La direcció d'ordenació que es fa servir, és la direcció que ens retorna la funció GetSortDirection. Finalment comprova el valor de l'atribut FlagImage i depenent del valor, crida al procediment AddSortImage o no.
- * *GridViewRowCommand(sender,e)*: aquest procediment és el responsable de capturar l'esdeveniment que es produeix quan es fa clic a algun dels botons que es troben al control GridViewForm. Un cop s'ha capturat, es comprova que el boto que ha desencadenat l'esdeveniment, sigui el boto que obra el Pop-Up, si és aquest, busca la fila on es troba, tot seguit es concatenen els ítems que es troben a la fila i a les columnes que indica l'atribut colsDetall, a l'atribut popupURL, per després poder passar els ítems com a variables que es passen per la URL¹⁶. Finalment es crida al procediment implementat amb codi JavaScript, passant-li com argument l'atribut popupURL.
- * *GridPageIndex(sender,e)*: procediment que captura l'esdeveniment que es produeix quan es clica un dels botons de paginació. El que fa, és assignar l'índex de la nova pàgina que s'ha de mostrar i tornar a enllaçar el control amb l'origen de dades. Per acabar, comprova si s'ha afegit l'imatge d'ordenació i si és així, crida al procediment AddSortImage.

2.5.2 JavaScript:

Amb aquest llenguatge s'han implementat diversos procediments, com per exemple validar les dades introduïdes per l'usuari o mostrar l'ajuda. S'ha utilitzat

¹⁶Uniform Resource Locator

el JavaScript per tal de que alguns procediments corrin al costat del navegador i d'aquesta forma evitar el PostBack del VB.NET, per tal d'alliberar la càrrega del servidor.

- scwshow(scwEle,scwSource): procediment encarregat de crear i mostrar un calendari per pantalla i introduir la data seleccionada per l'usuari en un TextBox, l'identificador d'aquest TextBox es passa al procediment com un argument.
- validarData(sender,args): aquest procediment és el responsable de validar la data introduïda per un usuari. Se li passa dos arguments, el sender, que és el control CustomValidator de la classe Data i l'altre que és la pàgina que conté el control Data. Primerament es busca el camp que és vol validar mitjançant el paràmetre sender, un cop el tenim, fem la validació utilitzant les expressions regulars de JavaScript i si la validació és incorrecta assignem el valor False al paràmetre args. Aquest procediment valida que la data només contingui dígit, que tingui el format dd-mm-aaaa, que el dia sigui un interval que va de 1-31 i el mes un interval que va de 1-12.
- validarHora(sender,args): aquest procediment valida l'hora introduïda per l'usuari, és molt similar a l'anterior, l'única diferència es troba amb l'expressió regular que s'utilitza per validar. L'hora ha d'estar formada per dígit, els seu format ha de ser hh:mm, les hores han d'anar d'un interval del 0-23 i els minuts del 0-59.
- validarNIF(sender,args): procediment que valida el número de NIF introduït per l'usuari. El funcionament és igual que en els dos procediments anteriors i l'única diferència és l'expressió regular utilitzada. El format del número de NIF ha de ser una cadena de 8 dígit.
- validarNIE(sender,args): procediment que valida el número de NIE, l'única diferència respecte als procediments anteriors és l'expressió regular utilitzada. El format del número del NIE ha de ser una cadena de 8 caràcters, on el primer caràcter pot ser, X,Y o Z i aquest va seguit de set dígit més.

- validarLletraNIF(sender,args): procediment que valida la lletra del NIF introduïda per l'usuari. Se li passa els mateixos arguments que als procediments anteriors. El primer que fa aquest procediment, és buscar el camp que conté la lletra mitjançant l'argument sender, a partir d'aquest camp, es busca el camp que conté el número de NIF utilitzant l'interfície de programació d'aplicacions DOM¹⁷. Un cop es té el número de document, es valida mitjançant les expressions regulars. Si la validació es correcta, es crida a la funció calcularLletra, que és la responsable de calcular si la lletra és correcta i retornar el resultat, depenent del resultat retornat, se li assigna a l'argument args el valor True o False.
- validarLletraNIE(sender,args): procediment que valida la lletra del NIE introduïda per l'usuari. També se li passa els mateixos arguments que als procediments anteriors. El seu funcionament és similar al del mètode anterior, primerament es busca el camp que conté la lletra del NIE, tot seguit es busca el camp que conté el número del NIE i es valida utilitzant una expressió regular. Si la validació és correcta es substitueix la lletra X per un 0, la lletra Y per un 1 i la lletra Z per un 2, després es crida a la funció calcularLletra i depenent del que torni, se li assigna el valor True o False a l'argument args.
- calcularLletra(dni,lletra): funció que valida si la lletra introduïda en el NIF o el NIE es correcta i retorna True o False depenent de la validació. A la funció se li passa dos arguments, el dni, que és el número de NIF o el número de NIE substituint les lletres X,Y,Z, pels seus valors corresponents. L'altre és la lletra del NIE o del NIF que es vol comprovar. L'algorisme implementat és el que es pot observar a algorithm 1.
- validarTel(sender,args): procediment que comprova que el número de telèfon introduït per un usuari sigui una cadena de nou dígit. Se li passa dos arguments, el sender, que és el control CustomValidator de la classe Telefon i el segon argument, és la pàgina que conté el control Telefon.

¹⁷Model d'Objectes del Document

Algorithm 1 Algorisme que comprova si la lletra del NIF o el NIE és correcta

```

taula=[T,R,W,A,G,M,Y,F,P,D,X,B,N,J,Z,S,Q,V,H,L,C,K,E]
index=dni mod 23
elementVerificador=taula[index]
if elementVerificador  $\neq$  lletra then
    Retornar False
end if
Retornar True

```

Primerament, mitjançant l'argument sender, es busca el camp que es vol validar. Un cop tenim el camp, fem la validació utilitzant les expressions regulars de JavaScript i si la validació és incorrecta, assignem el valor False a l'argument args.

- validarMail(sender,args): aquest procediment és el responsable de validar la sintaxi del correu electrònic introduït per un usuari. El funcionament és el mateix que el procediment anterior, amb l'única diferència de l'expressió regular utilitzada.
- validarNumCompte(sender,args): procediment responsable de validar el número de compte introduït per l'usuari i comprovar que el codi de control sigui correcte. Se li passa els mateixos arguments que al procediment anterior. Primer es busca el camp que conté els deu dígitos pertinents al número de compte, un cop s'ha trobat aquest camp, es busca els camps que contenen els dígitos respectius a l'entitat, la sucursal i el codi de control, utilitzant l'interfície de programació d'aplicacions DOM. Quan s'han trobat tots els camps, es concatenen i es valida la concatenació mitjançant una expressió regular que és una cadena formada per 20 dígitos. Després es crida a la funció codiControl, per tal que calculi si el codi de control és correcte i ens retorni True o False. Si la validació o el codi de control són incorrectes se li assigna a l'argument args el valor False.
- codiControl(numero): aquesta funció és la responsable de calcular el codi de control del número de compte, si el codi de control és correcte retorna True i sinó False. Com es pot observar, se li passa un argument, que és la

concatenació dels quatre camps del número de compte, és a dir, els quatre dígitos de l'entitat, més els quatre de la sucursal, més els dos del codi de control, més els deu del compte. La funció implementa l'algorisme que es pot veure al algorithm 2.

Algorithm 2 Algorisme que comprova si el codi de control del número bancari és correcte

```

pesos=[6,3,7,9,10,5,8,4,2,1]
suma=0
for  $i = 7$  to  $i > -1$  step  $i = -1$  do
    suma=suma + numero[i]*pesos[7-i]
end for
dc=11 - (suma mod 11)
if dc = 11 then
    dc=0
end if
if dc = 10 then
    dc=1
end if
for  $i = 19$  to  $i > 9$  step  $i = -1$  do
    suma=suma + numero[i]*pesos[19-i]
end for
dc2=11 - (suma mod 11)
if dc2 = 11 then
    dc2=0
end if
if dc2 = 10 then
    dc2=1
end if
if dc  $\neq$  numero[8] then
    Retornar False
end if
if dc2  $\neq$  numero[9] then
    Retornar False
end if
Retornar True

```

- modificaHora(id,inc): procediment que ens permet incrementar o decrementar l'hora o els minuts del control Hora. Se li passa dos arguments,

un és l'identificador del camp que conté l'hora i l'altre és un booleà que indica si es vol incrementar o decrementar. Primerament, aquest procediment, comprova que hi hagi una hora introduïda, sinó hi és, afegeix l'hora del sistema. Després busca la posició del cursor dins del TextBox, si es troba a les hores, es modifica les hores i sinó els minuts. Un cop s'ha incrementat o decrementat l'hora o els minuts, es torna a posicionar el cursor a la posició que es trobava abans d'entrar amb aquest procediment. Per manipular les hores i els minuts s'ha utilitzat l'objecte Date de JavaScript.

- showHelp(id,missatge): mètode que mostra l'ajuda d'un control, se li passa dos arguments, l'identificador del control que es vol que es mostri l'ajuda i el missatge que s'ha de mostrar per pantalla. Aquest procediment, el que fa, és crear un TextBox només de lectura, afegir-li el missatge que s'ha passat com argument i posicionar el TextBox a la pantalla a partir de les coordenades del punter del ratolí. Després, mitjançant l'interfície de programació d'aplicacions DOM es busca el pare d'on penja el control i s'afegeix el TextBox al node pare.
- deleteHelp(id): procediment que amaga l'ajuda que es mostra per pantalla. Se li passa un sol argument, que és l'identificador del control del qual es vol amagar l'ajuda. Per amagar l'ajuda, s'esborra del node pare l'element TextBox creat al procediment anterior.
- openPopup(url): procediment que obra una finestra emergent, mitjançant el mètode window.open i passant la variable url com un paràmetre d'aquest mètode. La variable url és una variable que s'ha passat com argument al procediment openPopup i és la direcció de la nova pàgina que es vol obrir com a finestra emergent.

2.5.3 CSS

Aquest llenguatge s'ha utilitzat per definir la presentació dels nous controls implementats en el framework. El document CSS està dividit amb classes, on

cada una d'aquestes classes té el seu propi estil.

- webtable: aquesta classe dóna un estil al control TableForm. Li assigna un tipus de lletra, una mida als texts, alinea el text a l'esquerra i dóna un color i una amplada als bordes de la taula. Als elements, com per exemple, el TextBox i els botons que es troben dins de la taula, també els hi dóna un color i una amplada i defineix una separació entre els diferents elements.
- imhelp: aquesta classe dóna una amplada i una alçada a l'imatge d'ajuda dels diferents controls.
- webpanelTitle: dóna un estil a la part on es troba el títol del control PanelForm. El que fa, és assignar una amplada a la part del títol, assignar un color i una amplada als bordes, establir un color de fons i alinear el text a l'esquerra.
- webpanelTitleImage: classe responsable de donar un estil a l'imatge que hi ha a la part del títol del control PanelForm. Assigna un espai entre el borde i el contingut de l'element i posiciona l'imatge a l'esquerra del títol.
- webpanelTitleText: és la classe responsable de donar un estil al text del títol del control PanelForm. Al títol li assigna un color, una mida i un tipus de lletra.
- webpanelBody: responsable de donar un estil al cos del control PanelForm. Assigna una amplada, un tipus de lletra i la mida de la lletra, a més a més, d'alinear el text al centre.
- boton: aquesta classe dóna un estil al control boto. Li assigna un tipus de lletra i una mida al text, un color als bordes dels botons, un color de fons, un marge entre els bordes i el contingut de l'element i posiciona els elements a l'esquerra dels elements adjacents.
- barraBotons: classe responsable de donar un estil al control PanelBoton. El que fa la classe és assignar una amplada al control, un color de fons, una amplada als bordes i un color als bordes.

- gridviewform: classe que dóna un estil al control GridViewForm. Assigna un tipus de lletra, una mida de lletra, un color de fons i una amplada als bordes del control.

2.6 Cas pràctic

En aquest apartat s'explica com s'ha fet una petita aplicació de gestió Web utilitzant les eines implementades en el framework, per tal de fer una demostració del funcionament de les noves eines. L'aplicació desenvolupada és una petita aplicació per tal de que els treballadors de l'empresa per la qual es fa aquest projecte puguin notificar el motiu i la data de la seva absència.

Aquesta aplicació conté una base de dades implementada amb SQL SERVER 2005 EXPRESS EDITION, per tal de guardar l'informació corresponent als treballadors de l'empresa, als motius i a les dates de les seves absències, en la figura 2.17 es pot veure el disseny de la base de dades.

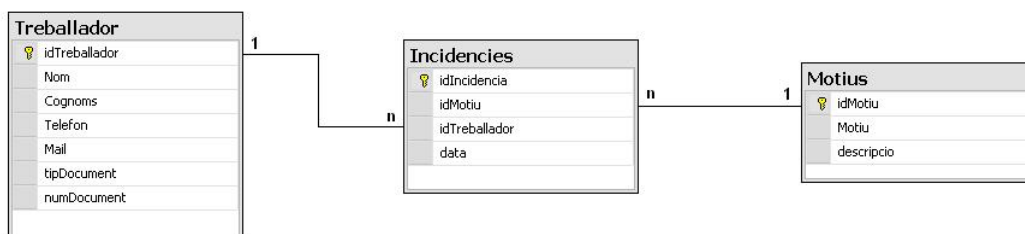


Figura 2.17: Disseny de la base de dades de l'aplicació corresponent al cas pràctic

La base de dades està formada per tres taules.

- Treballador: en aquesta taula es guarda l'informació dels treballadors de l'empresa. Cada element d'aquesta taula pot tenir associat un número indeterminat d'elements de la taula Incidencies.
- Incidencies: aquesta taula és la responsable de guardar l'informació de totes les absències dels treballadors.

- Motius: aquesta taula guarda els diferents motius per els quals un treballador pot estar absent a l'empresa. Cada element de la taula té associat un número indeterminat d'elements de la taula Incidencies.

L'aplicació permet fer tres operacions, afegir un treballador, afegir una absència i consultar les absències dels treballadors. A la figura 2.18 es pot veure una captura de pantalla de la primera pàgina de l'aplicació desenvolupada, aquesta, és un menú que et permet seleccionar l'operació a realitzar.



Figura 2.18: Captura de pantalla del menú de l'aplicació desenvolupada en el cas pràctic

L'operació corresponent a la d'afegir un treballador, et mostra una pàgina amb un formulari que permet a l'usuari introduir les dades corresponents a l'informació d'un empleat de l'empresa. A la figura 2.19 es veu la captura de pantalla de la pàgina corresponent amb aquesta operació. Quan es clica el boto Afegir es comprova que s'han introduït les dades en tots els camps i que són correctes, si és així s'afegeix l'informació a la taula Treballador de la base de dades.

A la figura 2.20 es pot observar una captura de pantalla de la pàgina corresponent a l'operació d'afegir una absència. Aquesta, conté un formulari per introduir les dades corresponents a una absència, si s'han introduït totes les dades i a més a més són correctes, quan es clica el boto afegir guarda l'informació a la taula Incidencies de la base de dades.

Si al menú es selecciona l'operació de realitzar una consulta, et redirigeix a una pàgina que conté un GridViewForm que mostra tots els treballadors de l'empresa. Aquest, et permet seleccionar el treballador del qual es vol consultar les absències. A la figura 2.21 es pot veure una captura de pantalla.

Figura 2.19: Captura de pantalla de la pàgina corresponent a l'operació d'afegir un treballador de l'aplicació desenvolupada en el cas pràctic

Treballador:	Nom	Cognoms
Seleccionar	Marc	Gutierrez Lopez
Seleccionar	Bernat	Sole Rodriguez
Seleccionar	Gerard	Jimenez Serrano
Seleccionar	Marta	Fuentes Ripoll
Seleccionar	Joan	Hernandez Sanchez
Seleccionar	Anna	Capdevila Ros
Seleccionar	Maria	Antunez Villoria
Seleccionar	Esther	Quesada Gili
Seleccionar	Jesus	Busquets Solsona
Seleccionar	Andreu	Seall Ribes

Figura 2.20: Captura de pantalla de la pàgina corresponent a l'operació d'afegir una absència de l'aplicació desenvolupada en el cas pràctic

A la figura 2.22 es pot veure el Pop-Up que apareix quan es selecciona el treballador del qual es vol consultar les seves absències. Aquest Pop-Up està format per un GridViewForm que mostra tota l'informació respectiva a les absències d'un treballador en concret.

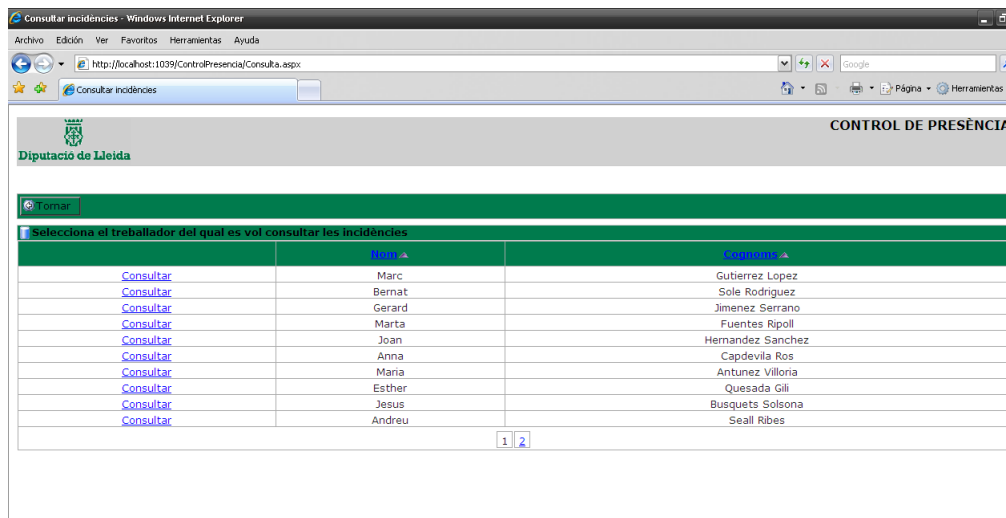


Figura 2.21: Captura de pantalla de la pàgina corresponent a l'operació de consultar absències de l'aplicació desenvolupada en el cas pràctic

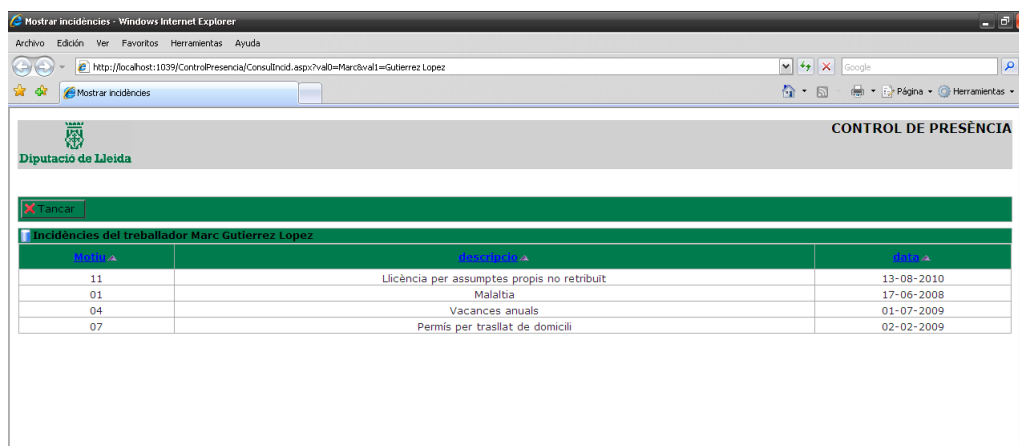


Figura 2.22: Captura de pantalla del Pop-Up que es desplega quan es selecciona el treballador del qual es vol consultar les absències

En aquesta aplicació s'han utilitzat diverses eines del framework, com per exemple, els controls PanelForm, TableForm, Boto, PanelBoton, LabelText, Telefon, eMail, DocumentIdentificatiu, Data i GridViewForm.

Capítol 3

Conclusions

Al llarg de la realització d'un framework, per tal que els programadors puguin desenvolupar aplicacions de gestió Web amb la plataforma de desenvolupament .NET, m'he vist obligat de treballar amb aquesta plataforma. Degut al treball que he fet amb aquesta plataforma he assolit un coneixement exhaustiu d'ella, per tant es pot dir que el primer objectiu ha estat complert. Una de les dificultats amb que m'he trobat a l'hora de programar ha estat amb elPostBack, ja que per exemple, es canviava la posició del cursor dins del control TextBox que forma part del control Hora.

En el framework s'ha implementat noves eines o controls, tant per codificar i optimitzar el temps de desenvolupament de la capa de presentació, com per facilitar l'enllaç al model de dades dins de l'entorn de desenvolupament. Després s'han utilitzat aquestes noves eines juntament amb els controls que proporciona la plataforma .NET, per tal de realitzar una petita aplicació de gestió Web, amb l'objectiu de fer una demostració dels elements desenvolupats en el framework i així contrastar que amb l'ajuda d'un framework s'optimitza i es facilita la tasca de programar per part dels programadors d'una empresa. Per tant es pot afirmar que tots els objectius han estat assolits.

Una altra de les conclusions extretes, es que en els formularis de les aplicacions de gestió Web de l'empresa per la qual s'ha fet el projecte acostumen a utilitzar els mateixos controls, per aquesta raó era interessant realitzar un framework amb

un conjunt d'elements implementats, ja que serien de gran utilitat en futures aplicacions de gestió Web.

Tal com s'ha pogut comprovar amb la petita aplicació que s'ha desenvolupat, per tal d'utilitzar-la com a demostració de les eines del framework, les noves eines implementades poden ser utilitzades juntament amb els controls que ofereix la plataforma .NET sense cap tipus de problema, d'aquesta forma se li dóna més llibertat al programador a l'hora de desenvolupar noves aplicacions, ja que no està restringit per les eines del framework desenvolupat.

Degut a la manera de com s'ha dissenyat el nou framework es pot ampliar amb la definició i l'implementació de noves eines sense cap tipus de problema. Una de les millores que es podria fer, és definir una nova eina, que és un panell que dóna la possibilitat de que només es produeixi el refresc en el panell i no en tota la pàgina, d'aquesta manera es podria reduir el treball realitzat per els servidors. Es podria realitzar el desenvolupament d'aquest panell utilitzant les funcionalitats ASP.NET AJAX¹.

L'ús del JavaScript en aplicacions Web desenvolupades amb la plataforma .NET és molt útil, ja que es poden fer moltes coses sense la necessitat de que es produeixi un refresc de pantalla o PostBack i d'aquesta manera s'aconsegueix més velocitat degut a que algunes parts de la nova aplicació s'executen al costat del client, al executar-se al costat del client, no ho fan al costat del servidor i així s'evita un camí d'anada i tornada. A més a més, la plataforma .NET té eines que faciliten l'integració del JavaScript en aplicacions .NET.

La part més problemàtica del projecte va ser implementar la validació de dades introduïdes per l'usuari. Aquesta part va donar problemes, ja que des d'un principi es volia fer la validació del costat del client, però per complicacions en el JavaScript a l'hora d'evitar l'enviament de les dades del formulari si eren incorrectes o com perquè tots el controls Button de la plataforma .NET produeixen l'esdeveniment submit, es va replantejar l'opció d'utilitzar algun dels controls de validació que proporciona la plataforma .NET. El problema d'aquests controls és que s'executen al costat del servidor, i el fet de realitzar la validació del costat

¹conjunt d'extensions per ASP.NET desenvolupat per Microsoft per implementar la funcionalitat de AJAX, anteriorment s'anomenava Atlas

del servidor era contrari a l'idea inicial. Finalment, a base de molta recerca es va trobar el control de la plataforma .NET, CustomValidator, que permet realitzar validacions tant al costat del servidor com del client d'una manera força senzilla per al programador.

Degut a que l'ús de la plantilla CSS que s'ha implementat per donar un estil a les eines del framework pot ser sempre la mateixa, es pot aconseguir que les noves aplicacions de gestió Web desenvolupades per part de l'empresa sempre tinguin una aparença molt similar.

Apèndix A

Eina de gestió d'una escola

És una petita eina de gestió d'una escola, on es guarda l'informació dels alumnes en una base de dades, l'informació guardada és les dades de contacte de l'alumne i el seu expedient acadèmic. Aquesta petita eina està destinada a dos tipus d'usuaris, un administrador, que és el que pot realitzar accions d'administració, com per exemple, afegir, eliminar un alumne, posar notes i modificar les dades de contacte de l'alumne o les seves notes. L'altra tipus d'usuari només permet consultar l'expedient acadèmic d'un alumne.

- Programari utilitzat
 - Microsoft Visual Studio 2005
 - Microsoft SQL Server Express 2005
- Llenguatge utilitzat
 - C#
 - ASP.NET

- Funcionament de l'aplicació

Tal com es veu a la figura A.1, la pàgina d'inici dona l'opció d'entrar com administrador o com un usuari normal. Si vols entrar com administrador et demana que t'identifiquis introduint el nom i la contrasenya, es pot observar a la figura A.2.

Si t'identifiques com administrador, un cop identificat, es pot escollir quina operació es vol realitzar, es pot afegir un alumne, posar una nota i modificar les dades personals o l'expedient acadèmic d'un alumne en concret. Es pot veure una captura de pantalla a la figura A.3.

Quan l'administrador selecciona l'operació a realitzar, l'aplicació et redirigeix a la pàgina corresponent a l'operació. Aquesta nova pàgina et demana que s'entri per teclat les dades necessàries per realitzar l'acció determinada. A les figures A.4, A.5, A.6, A.7 es pot veure les diferents captures de pantalla de cada una de les pàgines corresponent a l'operació que es vol dur a terme.

En el cas que inicis la sessió com un usuari normal, apareix una nova capa, en la que s'ha d'introduir el DNI de l'alumne i les assignatures de les qual es vol consultar la nota de l'expedient acadèmic. Es pot observar a la captura de pantalla que es troba a la figura A.8.

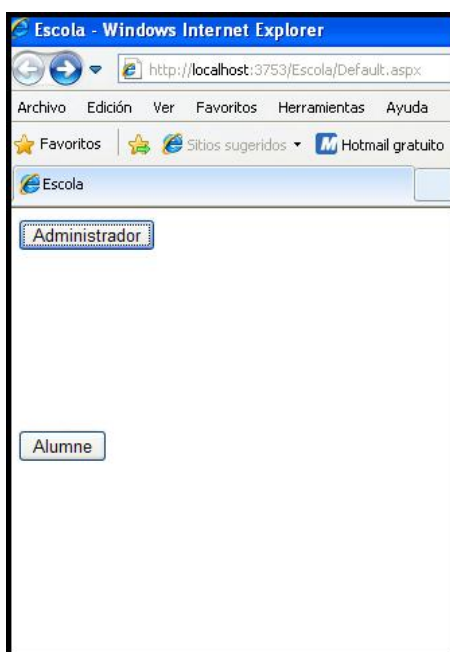


Figura A.1: Captura de pantalla on un usuari pot elegir entre el perfil d'administrador o usuari normal

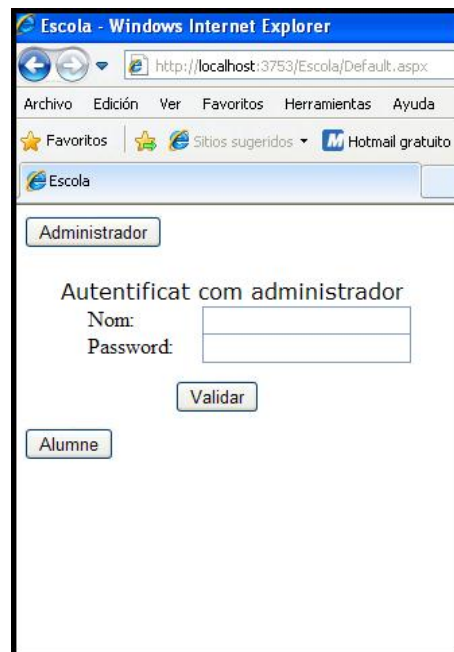


Figura A.2: Captura de pantalla on demana que l'administrador s'identifiqui mitjançant un nom i una contrasenya

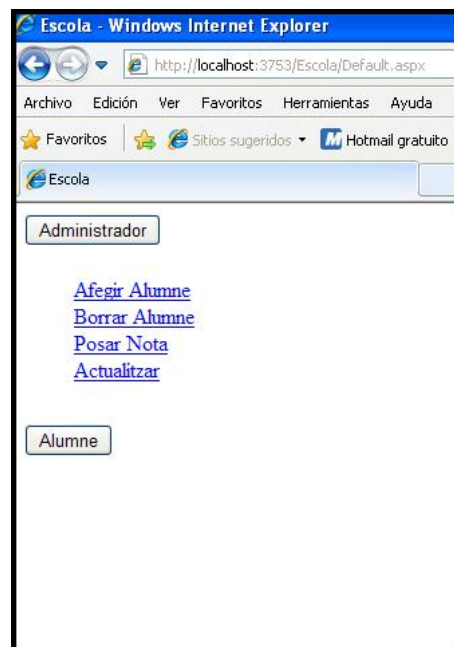
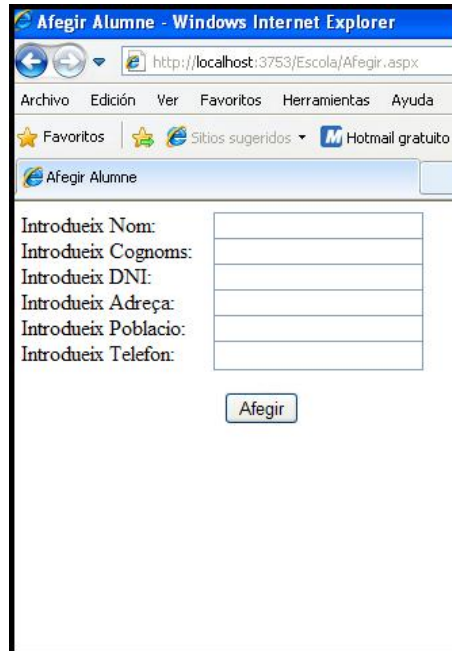


Figura A.3: Captura de pantalla on un administrador pot elegir l'operació a realitzar

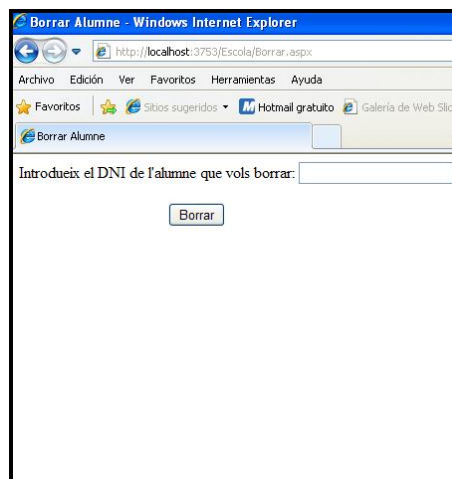


A screenshot of a web browser window titled "Afegir Alumne - Windows Internet Explorer". The address bar shows "http://localhost:3753/Escola/Afegir.aspx". The menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The toolbar shows "Favoritos", "Sitios sugeridos", "Hotmail gratuito", and a search bar. The page content is titled "Afegir Alumne" and contains a form with the following labels and input fields:

- Introdueix Nom:
- Introdueix Cognoms:
- Introdueix DNI:
- Introdueix Adreça:
- Introdueix Població:
- Introdueix Telefon:

Below the form is a button labeled "Afegir".

Figura A.4: Captura de pantalla que permet afegir un alumne a la base de dades



A screenshot of a web browser window titled "Borrar Alumne - Windows Internet Explorer". The address bar shows "http://localhost:3753/Escola/Borrar.aspx". The menu bar includes "Archivo", "Edición", "Ver", "Favoritos", "Herramientas", and "Ayuda". The toolbar shows "Favoritos", "Sitios sugeridos", "Hotmail gratuito", and "Galería de Web Slices". The page content is titled "Borrar Alumne" and contains a form with the following label and input field:

- Introdueix el DNI de l'alumne que vols borrar:

Below the form is a button labeled "Borrar".

Figura A.5: Captura de pantalla que permet eliminar un alumne existent a la base de dades

Figura A.6: Captura de pantalla que permet posar notes d'un alumne existent a la base de dades

Figura A.7: Captura de pantalla que permet modificar algun dels paràmetres d'un alumne existent a la base de dades

Assignatura	Nota
Llengua Catalana	6
Llengua Castellana	7.5
Matemàtiques	6
Ciències Socials	8
Ciències Naturals	8.5
Llengua Inglesa	4

Figura A.8: Captura de pantalla que mostra l'expedient acadèmic d'un alumne en concret

Apèndix B

Servei Web

És un servei Web que et permet realitzar les quatre operacions matemàtiques bàsiques: sumar, restar, multiplicar i dividir. Per tal de realitzar una operació, només és necessari introduir dos números i invocar el mètode desitjat.

- Programari utilitzat
 - Microsoft Visual Studio 2005
- Llenguatge utilitzat
 - C#

- Funcionament de l'aplicació

El funcionament d'aquesta aplicació és molt senzill, només cal introduir dos números i invocar el mètode corresponent a l'operació matemàtica que es vol realitzar. Es poden veure les diferents captures de pantalla del servei Web a les figures B.1, B.2, B.3, B.4 i B.5.

El resultats retornats per cada una de les operacions es poden veure a la figura B.6.

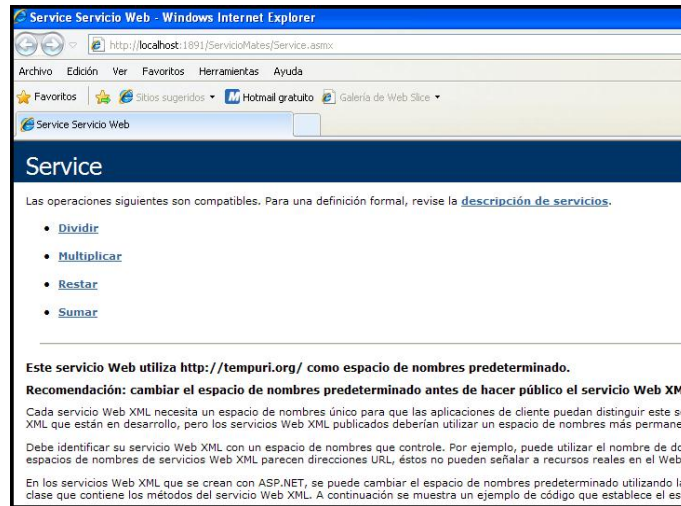


Figura B.1: Captura de pantalla que mostra els diferents mètodes que es poden invocar

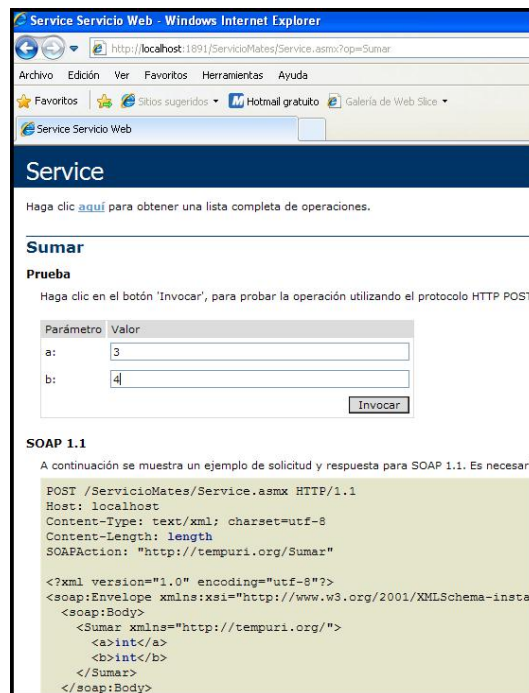


Figura B.2: Captura de pantalla que mostra l'operació de sumar

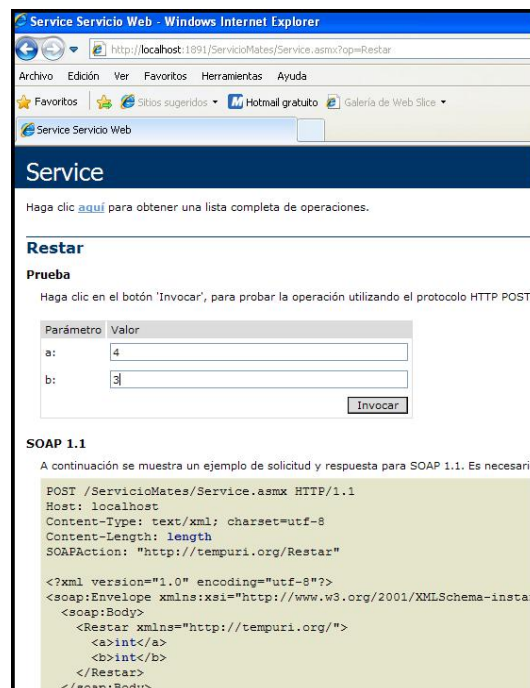


Figura B.3: Captura de pantalla que mostra l'operació de restar

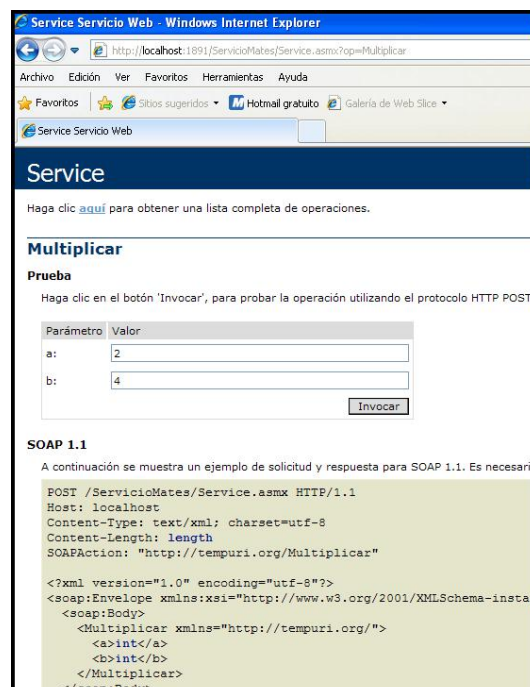
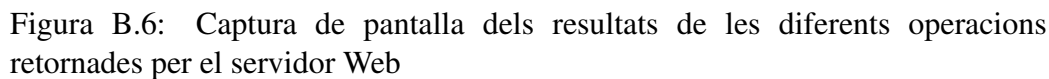
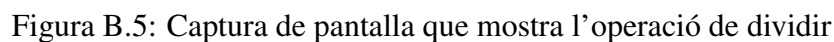


Figura B.4: Captura de pantalla que mostra l'operació de multiplicar



Apèndix C

Aplicació Web utilitzant la llibreria d'empresa X

És una petit portal on es poden consultar les estadístiques de diferents jugadors de futbol de diferents equips en diferents temporades. Aquest portal ha estat implementat amb l'ajuda d'un framework desenvolupat per una empresa. En aquest portal hi ha dos tipus de permisos d'usuaris, el permís d'administrador, que permet afegir, eliminar, modificar i consultar dades. L'altre permís és el d'usuari, que només permet realitzar consultes.

- Programari utilitzat
 - Microsoft Visual Studio 2005
 - Microsoft SQL Server Express 2005
- Llenguatge utilitzat
 - VB.NET
 - ASP.NET
- Funcionament de l'aplicació

A la figura C.1 es pot veure una captura de pantalla de la pàgina d'inici, aquesta permet iniciar la sessió com un usuari ja registrat o donar d'alta un

76 APÈNDIX C. APLICACIÓ WEB UTILITZANT LA LLIBRERIA D'EMPRESA X

usuari nou. Si es vol donar d'alta un nou usuari, el portal et redirigeix a la pàgina de la figura C.2.

Si s'inicia la sessió com un usuari registrat, l'eina et redirigeix a una nova pàgina on es pot veure una captura de pantalla a la figura C.3. Aquesta nova pàgina, que és el menú, permet seleccionar les operacions de consultar, afegir, eliminar o editar les dades. En el cas que la sessió s'hagi iniciat com un usuari amb permisos d'usuari, només permet realitzar l'operació de consultar.

A les figures C.4, C.5, C.6 i C.7 es pot veure les captures de pantalla de cada una de les diferents operacions que es poden realitzar. Tant l'operació d'afegir com eliminar, permeten afegir o eliminar un jugador o un equip.



Figura C.1: Captura de pantalla que mostra la pàgina d'inici del portal



Figura C.2: Captura de pantalla de la pàgina que permet donar d'alta un nou usuari



Figura C.3: Captura de pantalla de la pàgina del menú d'inici

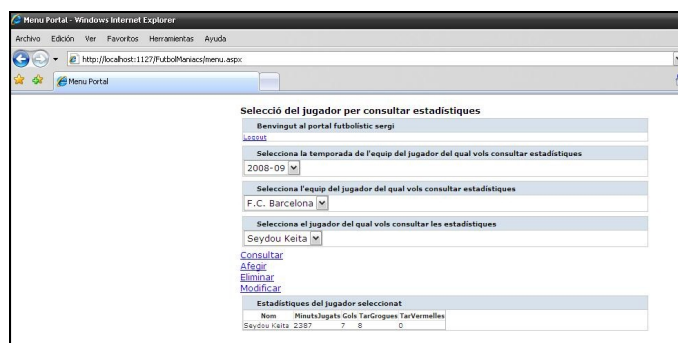


Figura C.4: Captura de pantalla de les dades retornades de l'operació consultar

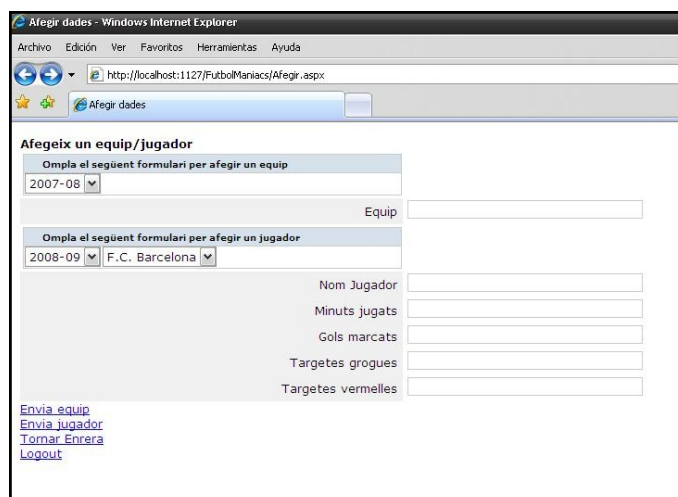


Figura C.5: Captura de pantalla que mostra la pàgina per afegir dades



Figura C.6: Captura de pantalla que mostra la pàgina per eliminar dades

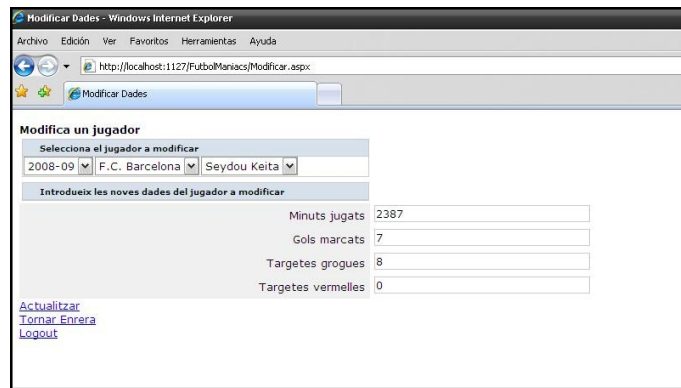


Figura C.7: Captura de pantalla que mostra la pàgina per modificar dades

Bibliografia

- [1] <http://msdn.microsoft.com/es-es/netframework/default.aspx>
Últim accés: 29-3-2010
- [2] <http://msdn.microsoft.com/es-es/aa336522.aspx>
Últim accés: 26-2-2010
- [3] <http://msdn.microsoft.com/es-es/vbasic/default.aspx>
Últim accés: 3-8-2010
- [4] <http://msdn.microsoft.com/es-es/vstudio/default.aspx>
Últim accés: 29-3-2010
- [5] <http://msdn.microsoft.com/es-es/sqlserver/default.aspx>
Últim accés: 3-8-2010
- [6] <http://www.w3schools.com/js/default.asp>
Últim accés: 3-8-2010
- [7] <http://www.w3schools.com/css/>
Últim accés: 3-8-2010
- [8] <http://msdn.microsoft.com/es-es/library/tt9b3d0f%28v=VS.80%29.aspx>
Últim accés: 3-8-2010
- [9] <http://msdn.microsoft.com/es-es/library/a63e36w2%28v=VS.80%29.aspx>
Últim accés: 3-8-2010

- [10] <http://msdn.microsoft.com/es-es/library/ms227437%28v=VS.80%29.aspx>
Últim accès: 22-1-2010
- [11] <http://msdn.microsoft.com/es-es/library/debza5t0%28v=VS.80%29.aspx>
Últim accès: 22-1-2010
- [12] <http://msdn.microsoft.com/es-es/library/w25ef46s%28v=VS.80%29.aspx>
Últim accès: 22-1-2010
- [13] <http://msdn.microsoft.com/es-es/library/d51ttbhx%28v=VS.80%29.aspx>
Últim accès: 22-1-2010
- [14] <http://www.djangoproject.com>
Últim accès: 26-2-2010

Firmat: Sergi Pedra Pagès

Bellaterra, Setembre de 2010

Resum

Hi ha moltes empreses que fan aplicacions de gestió Web. Una d'aquestes empreses, és per la que s'ha fet el projecte. El problema és que tenen molts projectes per desenvolupar, per tant van decidir d'implementar un conjunt d'eines amb l'objectiu de facilitar l'implementació de les aplicacions als programadors. S'ha desenvolupat una llibreria .dll, que conté aquestes noves eines. També s'ha utilitzat codi JavaScript i CSS per tal d'implementar alguna funcionalitat i donar una presentació amb aquestes noves eines.

Resumen

Hay muchas empresas que hacen aplicaciones de gestión Web. Una de estas empresas, es por la que se ha hecho el proyecto. El problema es que tienen muchos proyectos para desarrollar, por lo tanto decidieron implementar un conjunto de herramientas con el objetivo de facilitar la implementación de las aplicaciones a los programadores. Se ha desarrollado una librería .dll, que contiene estas nuevas herramientas. También se ha utilizado código JavaScript y CSS para implementar alguna funcionalidad y dar una presentación a estas nuevas herramientas.

Abstract

There are many companies make Web management application, one of these companies, is the company for that I make the project. The problem is that have many projects to develop, therefore they decided to implement a toolkits with the objective of facilitating the implementation the applications to programmers. Has been developed a .dll library, wich contains these new toolkits. Also has been used JavaScript and CSS code to implements some functionality and give a presentation to these new toolkits.